



UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE FÍSICA

TRABALHO DE LICENCIATURA

CURSO DE LICENCIATURA EM METEOROLOGIA

Previsão de Trajectória de Ciclones Tropicais na Bacia do Sudoeste do Oceano Índico Baseado em Modelos de Aprendizado Profundo

AUTOR:

NHAMPIMBE, LÚCIO HILÁRIO



UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE FÍSICA

TRABALHO DE LICENCIATURA

CURSO DE LICENCIATURA EM METEOROLOGIA

Previsão de Trajectória de Ciclones Tropicais na Bacia do Sudoeste do Oceano Índico Baseado em Modelos de Aprendizado Profundo

Supervisor:

Prof. Doutor Atanásio Manhique

Co-Supervisor:

Prof. Doutor Pietro Pinoli

Maputo, Janeiro de 2025

Previsão de Trajectória de Ciclones Tropicais no Sudoeste do Oceano Índico Baseado em Modelos de Aprendizado Profundo

Dedicatória

Este trabalho é dedicado a toda minha família, especialmente a minha falecida mãe Rabeca Lúcia Paruque, aos meus tios Fernando Bangane e Rosa D. dos Santos Paruque, e aos meus irmãos Michel M. Benjamim, Loide R. Manuel, Hilário J. Nhampimbe Júnior.

Previsão de Trajectória de Ciclones Tropicais no Sudoeste do Oceano Índico Baseado em Modelos de Aprendizado Profundo

Declaração de Honra

O presente trabalho foi completamente produzido e realizado com base no material que ao longo do mesmo se faz referência. As ideias originais nele espessas, são de inteira responsabilidade do autor.

Maputo, Janeiro de 2025

Autor

(Lúcio Hilário Nhampimbe)

Previsão de Trajectória de Ciclones Tropicais no Sudoeste do Oceano Índico Baseado em Modelos de Aprendizado Profundo

Agradecimentos

A materialização deste trabalho foi possível graças ao amor, carinho, críticas e encorajamento de familiares, amigos, colegas e professores. Portanto, agradeço :

- A Deus pela proteção, energia e saúde em toda minha vida;
- Aos meus pais Gonçalves Ndongane e Rabeca Lúcia Paruque por todo o amor e carinho que me proporcionaram durante os primeiros momentos da minha existência, graças aos quais, também, estou neste mundo curioso podendo explorar e contribuir;
- Aos meus Tios Fernando Bangane e Rosa David dos Santos Paruque, pela educação e apoio que sempre têm me proporcionado, assim como por serem figuras maternas e parterna para mim.
- Aos meus tios Gonçalves Lampião, Atália Lúcia e Lídia David dos Santos Parque pela educação, carinho e apoio que sempre têm me proporcionado;
- Aos meus queridos irmãos que têm apoiado a minha caminhada académica, especialmente o Michel M. Benjamin, Hilário Nhampimbe Júnior e Loide Manuel pelo apoio carinho, amor, conselhos de Pais que eles sempre têm me dado;
- Aos meus primos e Sobrinhos, Especialmente Gonçalves Mapossa, Manuel Mapossa, Fernando Bangane Júnior, Atália R. Mapossa, Marnela I. Mapossa, Edvaldo M. Mapossa, Fernando Gonçalves Júnior, Gonçalves A. Lampião Júnior, Fernando Gonçalves Mapossa, Rabeca T. Machava e Lutegardo Lampião pelos momentos de irmandade que passamos juntos durante a minha infância;
- Ao meu Supervisor Prof. Doutor Atanásio Manhique, pela orientação e encorajamento na realização deste trabalho, assim como por ser um ótimo professor;
- Ao meu Co-Supervisor Prof. Doutor Pietro Pinoli por me fazer acreditar na possibilidade da realização deste trabalho, pelo encorajamento, orientação, e por ter se disposto a me orientar desde a nossa primeira interação em 2024 durante o programa ICT4Dev da UEM;
- A todos os meus colegas desta caminhada, em especial aos colegas e amigos Arlindo Nhanguatata, Paulo Meque, Jéssica Malungane, Wilson Guambe, Nuhi Lazima e Eulália Chiau, por me terem dado força, apoio e direção, pois juntos partilhamos momentos inesquecíveis, quer fosse ‘matando’ ou ‘morrendo’ com cadeiras.
- Aos meus amigos de infância e algumas amigadas que fiz recentemente, especialmente Ritchasse Malhango e David Chiconela por terem me emprestado os seus computadores sempre que precisei, assim como pelos momentos de irmandade que vivemos recentemente, os meus amigos

Previsão de Trajectória de Ciclones Tropicais no Sudoeste do Oceano Índico Baseado em Modelos de Aprendizado Profundo

de longa data Carlos Ndzauane, Allan do Rosário, Paulo tembe, Iracema dos Santos e o grupo ‘A Malta’ por todos os momentos inesquecíveis que passámos e por directa ou indirectamente terem contribuído para que se tornasse possível a realização deste trabalho aqui vai o meu obrigado;

- A minha tia e vizinha Leonor Tembe, assim como a sua família, especialmente o seu filho António Linder pelos momentos inesquecíveis da minha infância, e também por ter sido a minha primeira explicadora e por continuar a ser alguém com posso contar até hoje;
- A Universidade Eduardo Mondlane e a todos os professores da UEM que participaram da minha formação e, por todo o conhecimento que me tem proporcionado;
- Ao Professor Steven Bruton da Universidade Washington por seu livro ‘*Data Driven science and engineering*’ e por suas inúmeras aulas sobre *machine learning* e ciência de dados gratuitamente disponíveis no YouTube que me ajudam imenso toda vez que preciso de aprender mais sobre o excitante campo de ciência de dados;
- Ao Professor Aurélien Géron por seu livro incrível ‘*Hands-on Machine Learning with Scikit-learn, keras, and TensorFlow*’ que foi um dos primeiro e mais impactantes contactos que tive com o aprendizado de máquinas;
- A Universidade de Stanford e Massachusetts Institute of Technology (MIT) pelas aulas incríveis sobre *deep learning* e *machine learning* que são de livre acesso e que ajudaram-me bastante a aprofundar os meus conhecimentos nas áreas de DL e ML;
- Aos incríveis membros da comunidade do Python por facilitarem o aprendizado da linguagem através das aulas e tutoriais gratuitamente disponíveis na Internet, espialmente ao Jason Brownlee pelos tutoriais incríveis e práticos;
- A toda gente conhecida, parente e amiga que directa ou indirectamente contribuíram para que se tornasse possível a realização deste trabalho, aqui vai o meu OBRIGADO;
- Por último e não menos importante, agradeço a essa força e energia que há dentro de mim pela garra e determinação na busca da materialização dos meus sonhos e projectos.

Previsão de Trajectória de Ciclones Tropicais no Sudoeste do Oceano Índico Baseado em Modelos de Aprendizado Profundo

Lista de Símbolos

Símbolos	Descrição
X	Matriz dos dados de entrada (variável independente, <i>input</i>)
Y	Matriz dos dados de saída (variável dependente, <i>output</i>)
Y'	Matriz dos dados dos <i>outputs</i> produzidos pelo modelo de previsão
x^{teste}, y^{teste}	Uma amostra (<i>Input e Outputs</i>) dos dados de teste
x^{trein}, y^{trein}	Uma amostra (<i>Input e Outputs</i>) dos dados de treinamento
MSE_{teste}	Erro Médio quadrado do modelo nos dados de teste
MSE_{trein}	Erro Médio quadrado do modelo nos dados de treinamento
W	Matriz dos coeficientes (<i>weights</i>) dos modelos
b	Matriz dos <i>bias</i> dos modelos
$L(w,b)$	<i>Loss</i> (custo) do modelo em função do w e b do modelo
$\sigma()$	Função de activação
$\delta_{lat}, \delta_{long}$	Varição de Latitude e Longitude
$P(y x)$	Probabilidade do evento y dado que o evento x ocorreu

Lista de Abreviaturas

Abreviaturas	Nome completo
CT	Ciclones Tropical
SOI	Sudoeste do Oceano Índico
INAM	Instituto Nacional de Meteorologia
UEM	Universidade Eduardo Mondlane
°S, °E	Graus Sul, Graus Este
IBTrACS	International Best Track Archive for Climate Stewardship
TSM	Temperatura da superfície do mar
ML	<i>Machine Learning</i>
DL	<i>Deep Learning</i>
MSE	<i>Mean Squared Error</i> (Erro Médio Quadrado)
CNN	<i>convolutional neural network</i> (Rede neural convolucional)
RNN	<i>Recurrent Neural Networks</i> (Redes Neurais Recursivas)
LSTM	<i>Long Short-Term Memory</i>
GRU	Gated Recurrent Unit
MAE	<i>Mean Absolute Error</i> (Error Médio Absoluto)
RMSE	<i>Root Mean Squared Error</i> (Raíz do Erro Médio Quadrado)
ED	Erro da distância
GD	Gradiente Descente (<i>Gradiente Descent</i>)
RFR	Random Forest Regressor
PNT	Previsão Numérica do Tempo
IA	Inteligência Artificial

Resumo

Os ciclones tropicais (CTs) são sistemas de tempestade intensos, de escala sinóptica, que se formam sobre oceanos tropicais e estão ligados a perigos meteorológicos catastróficos, incluindo ventos extremos, chuvas torrenciais, e ressacas, além de perdas econômicas. Gerar previsões rápidas e precisas das suas trajetórias é crucial para o desenvolvimento de estratégias de prevenção e mitigação. O movimento dos CTs é governado por interações complexas e não lineares entre variáveis atmosféricas, condições oceânicas e fluxos de circulação mais amplos, tornando sua previsão de trajetória um desafio intrínseco tanto para modelos dinâmicos quanto estatísticos. Este estudo utiliza o Aprendizado Profundo (Deep Learning, DL) —um subtipo de aprendizado de máquina que emprega redes neurais hierárquicas para extrair autonomamente padrões espaciotemporais de conjuntos de dados complexos—para desenvolver modelos preditivos das trajetórias de CTs nas próximas 6–72 horas sobre a Bacia do Sudoeste do Oceano Índico (SOI). Os movimentos dos CTs sobre o SOI, de 1993 a 2023, foram analisados com base em dados extraídos do banco de dados International Best Track Archive for Climate Stewardship (IBTrACS). A partir dos dados de entrada, foram selecionadas 11 variáveis relevantes que melhor explicam as mudanças na latitude e longitude dos CTs nas próximas 6–72 horas utilizando o algoritmo Random Forest Regressor. A seleção foi baseada nos índices de importância classificados das variáveis. Cinco modelos baseados em redes neurais recorrentes (RNN) foram desenvolvidos e treinados para prever as trajetórias dos CTs utilizando as últimas 24 horas de dados meteorológicos e de trajetória observada (das variáveis selecionadas) como entrada. A avaliação pós-treinamento, utilizando Erro Médio Absoluto (MAE), Raiz do Erro Quadrático Médio (RMSE) e erro de distância de trajetória, revelou um bom desempenho na simulação das trajetórias dos CTs no SOI pelos modelos Conv-GRU, Stacked-LSTM e Encoder-Decoder, com erros médios de distância variando de 19 a 160 km para as previsões de 6–24 horas nos dados de teste. Uma abordagem de conjunto (ensemble learning), combinando previsões dos modelos de melhor desempenho por meio de aprendizado em conjunto empilhado (Stacked Ensemble Learning), reduziu ainda mais os desvios nas trajetórias nas primeiras horas. A comparação com as previsões operacionais do Centro Regional de Ciclones Tropicais (La Réunion) demonstrou um bom alinhamento, particularmente nas previsões de curto prazo (<24h). Esses resultados destacam o potencial dos modelos baseados em DL para complementar os métodos numéricos e estatísticos existentes, oferecendo uma solução mais econômica e orientada por dados para a previsão operacional de trajetórias de ciclones tropicais na Bacia do SOI.

Palavras chaves: Ciclones Tropicais, Sudoeste do Oceano Índico (SOI), Previsão de trajetórias, Deep Learning.

Lista de Figuras

Figura 1: Bacia oceânica do SOI. -----	6
Figura 2: Imagem de satélite de Idai às 11h35 UTC de 14 de março de 2019. Fonte: (Kolstad, 2021)-	7
Figura 3: Secção vertical através de um ciclone tropical. -----	8
Figura 4: Mapa de localizações de CTs, com um quadro geral sentido das suas Trajetórias -----	11
Figura 5: Trajetórias dos sistemas tropicais sobre o SOI com legenda de intensidade baseado na escala da Tabela 3 dos dados IBTrACSS durante o período 1993–2023. -----	15
Figura 6: Programação Tradicional e Aprendizado de Máquinas. -----	16
Figura 7: Componentes de: Neurónio Artificial (a) e Neurónio Biológico (b). -----	25
Figura 8: <i>Rede Neural Feed Forward Totalmente Conectada.</i> -----	26
Figura 9: Processamento convolucional de imagem de um ciclone e Construção de Mapas de Características. -----	28
Figura 10: Camada convolucional. Fonte: Adaptado de (Zhang et al., 2023)-----	29
Figura 11: operações de pooling: esquerda - max pooling, direita - average pooling. -----	29
Figura 12: Estrutura geral de uma CNN -----	30
Figura 13: Neurónio Recursivo : um neurónio recursivo (esquerda) desenrolado através dos passos temporais (direita)-----	31
Figura 14: Unidade LSTM. -----	33
Figura 15: Conexões e Blocos Residuais -----	35
Figura 16: Método de deslizamento de janela usado para gerar os dados de X e Y de series temporais para o treinamento de modelos de Machine Learning. -----	36
Figura 17: arquitetura da rede stacked-lstm -----	37
Figura 18: Estrutura do Modelo BN-LSTM-----	38
Figura 19: Estrutura esquemática do modelo Skip-LSTM -----	38
Figura 20: Estrutura Esquemática do modelo Conv-GRU: a - Bloco Residual (ResBlock) e b - estrutura completa do modelo -----	39
Figura 21: Estrutura esquemática do Modelo ENCODER-DECODER-----	40
Figura 22: (a) o índice de importância das variáveis e (b) a pontuação OOB de diferentes combinações de variáveis.-----	49
Figura 23: Curvas de Aprendizagem: a - Modelo Stacked-LSTM, b - BN-LSTM-----	51
Figura 24: c - conv-GRU, d - Encoder - Decoder, e - SKIP-LSTM -----	52

Figura 25: Variação do erro médio da distância (entre posições reais e estimadas) dos 5 modelos de redes neurais propostos neste trabalho -----	56
Figura 26: Diagrama de caixa do erro médio da distância (entre posições reais e estimadas) dos 5 modelos (arquitecturas de redes neurais) propostos neste trabalho para fazer previsões de (δlat , δlon) de 6–72 horas (num intervalo de 6 horas). -----	57
Figura 27: Mapa da trajetória do CT Kenneth da temporada ciclónica de 2018-2019 sobre o SOI. ----	59
Figura 28: Mapa da trajetória do CT Eloise da temporada ciclónica de 2020-2021 sobre o SOI. -----	59
Figura 29: Mapa da trajetória do CT Eloise da temporada ciclónica de 2020-2021 sobre o SOI. -----	60
Figura 30: visualização gráfica do algoritmo GD -----	67

Lista de Tabelas

Tabela 1: Definição da escala de categorias de CTs com base no vento máximo sustentado de 1 minuto, medido na altura padrão do anemômetro de 10 m.....	9
Tabela 2: Escala de vento do furacão Saffir-Simpson - Descrição dos danos esperados (Stull, 2015) ...	9
Tabela 3: Escala usada pela Météo-France (para Hemisfério Sul, a oeste de 90 ° E) (Stull, 2015).....	10
Tabela 4: Requisito para <i>Ciclogénese</i>	11
Tabela 5: Ciclo de vida dos Ciclones Tropicais	13
Tabela 6: Hiperparâmetros de configuração dos modelos.....	41
Tabela 7: Lista de variáveis de Estudo	44
Tabela 8: Detalhes dos dados experimentais	49
Tabela 9: Erro de previsão dos modelos em termos de RMSE	53
Tabela 10: Erro de Previsão dos modelos em termos de MAE	53
Tabela 11: Comparação do Erro Médio da Distância produzidos por cada um dos modelos propostos no trabalho	54
Tabela 12: Pseudocódigo do GD	66

Índice

Dedicatória.....	i
Declaração de Honra	ii
Agradecimentos.....	iii
Lista de Símbolos	v
Lista de Abreviaturas	vi
Resumo.....	vii
Lista de Figuras	vi
Lista de Tabelas.....	viii
Capítulo 1 : Introdução.....	1
1.1. Objectivos	3
1.1.1. Objectivo Geral	3
1.1.2. Objectivos Específicos	3
1.2. Definição do Problema.....	3
1.3. Justificativa	4
1.4. Estrutura do Trabalho.....	5
Capítulo 2 : Revisão Bibliográfica.....	6
2.1. Área de Estudo	6
2.2. Ciclones Tropicais.....	7
2.2.1. Estrutura de Ciclones Tropicais	8
2.2.2. Intensidade e distribuição geográfica	9
2.2.3. Requisitos Para a Ciclogénese.....	11
2.2.4. Ciclo da vida dos Ciclones Tropicais	12
2.2.5. Climatologia das Trajectórias de Sistemas Tropicais sobre o SOI.....	14
2.2.6. Modelos Previsão de Trajetórias dos CTs	15
2.3. Fundamentos de Aprendizado de Máquinas	16
2.3.1. Descrição de Algoritmos de Aprendizado de Máquinas	16
2.3.2. Regressão Linear como um exemplo de Aprendizagem de Máquina	18
2.3.3. Aprendizado Profundo	22
2.3.4. Redes fundamentais de DL.....	24
2.3.5. Previsão de Séries Temporais com Métodos Supervisionados de Aprendizado de Máquinas.....	35
2.3.6. Modelos de Redes Neurais para previsão de trajectória de CTs	36

2.4. Estudos Relacionados.....	43
Capítulo 3 : Matérias e Métodos	44
3.1. Materiais.....	44
3.1.1. Coleção de Dados.....	44
3.2. Métodos.....	45
3.2.1. Pré-Processamento de Variáveis	45
3.2.2. Implementação de Redes Neurais Artificiais	46
3.2.3. Reconstrução de Trajectórias	46
3.2.4. Comparação de trajectórias e Contrução do Ensemble	47
Capítulo 4 : Resultados e Discussão	48
4.1. Seleção de Variáveis	48
4.2. Análise dos treinamentos	50
4.3. Análise da Performance dos modelos nos dados de Teste	53
4.3. Seleção de Modelos.....	57
4.4. Produção de Trajectórias.....	58
Capítulo 5 : Considerações finais e Recomendações	61
5.1. Conclusão.....	61
5.2. Recomendações.....	62
5.2.1. Recomendações para a Secção do Curso de Meteorologia da UEM.....	62
5.2.2. Recomendações para Próximas Pesquisas	62
5.2.3. Recomendações para o INAM	62
Bibliografia.....	63
Anexos.....	66

Capítulo 1 : Introdução

OS ciclones tropicais são sistemas de tempestade intensos, de escala sinóptica que se formam nos trópicos. Como todos os ciclones, os ciclones tropicais têm um centro de baixa pressão próximo do nível do mar. Além disso, os ventos de baixa altitude giram ciclonicamente (sentido horário no hemisfério sul) em torno da tempestade e em forma de espiral em direção ao centro da tempestade (Stull, 2015). Estes sistemas são geralmente acompanhados por tempestades que culminam em destruições e perdas de vidas humanas. Os desastres causados pelos ciclones tropicais são tipicamente inevitáveis, por tanto, prever as trajetórias dos ciclones tropicais ao longo do seu tempo vida é imprescindível para que as autoridades da região implementem estratégias de mitigação.

Sob a responsabilidade do Centro Meteorológico Regional Especializado La Réunion para alertas de sistemas tropicais, a bacia do Sudoeste do Oceano Índico(SOI) se estende do equador até $40^{\circ}S$ e da costa africana, de $(30^{\circ}E)$ até $90^{\circ}E$. E os sistemas tropicais que se formam nesta área podem afetar vários países ao longo da costa leste da África (Leroux, et al., 2018). Ao contrário da maioria das bacias em que todos os sistemas são denominados de ciclones tropicais independentemente da intensidade, no SOI, o termo "ciclone tropical" é reservado para descrever sistemas com velocidades máximas médias de vento de 10 minutos maiores ou iguais a $64 kt$ ($V_{MAX} \geq 32,9 m s^{-1}$). Os sistemas tropicais estão uniformemente distribuídos pela bacia de 35° a $95^{\circ} E$ (Leroux, et al., 2018) . Eles se movem principalmente para sudoeste, de acordo com a direção ambiental predominante (ventos alísios de leste) e com o efeito beta antes de se recurvarem para o sul ou sudeste para sua transição extratropical. Sobre o Oceano Índico aberto, os TCs que são equivalentes em intensidade a um furacão ou tufão e estão confinados principalmente à faixa de latitude $10^{\circ} - 25^{\circ}S$, ao norte do gradiente de frio-para-mais frio da temperatura da superfície do mar (Leroux, et al., 2018).

O movimento dos CTs é tipicamente complexo e não linear quanto a relação com outros parâmetros atmosféricos e oceânicos (Tian, et al., 2023) e, podem ser afectados por vários factores, incluindo o fluxo do vento, efeito beta, condições da estrutura adjacente, temperatura da superfície do mar, e circulação de meso-escala (Chan & Kepert, 2010). Portanto, prever a posições que o centro do CT pode tomar é um desafio. Actualmente, métodos de previsão para ciclones tropicais são principalmente divididos em duas categorias, com o primeiro método sendo a Previsão Numérica do Tempo (PNT) e outro método de previsão baseia-se em modelos estatísticos (DeMaria et al., 2005).

PNT consiste no cálculo da solução aproximada de equações diferenciais envolvendo variáveis do estado atmosférico quando as condições iniciais e de fronteira são conhecidas (ECWMF, 2019). Embora este

método seja preciso, as suas previsões dependem da qualidade das condições iniciais, e da disposição de supercomputadores. Por outro lado, o modelo estatístico é principalmente baseado nas relações históricas entre as características do ciclone e o seu movimento, mas normalmente não considera nenhum processo físico. A maioria dos modelos estatísticos para previsão CTs adoptam regressão linear, portanto é difícil para essas abordagens para resolver o problema da não linearidade na previsão da trajectória de ciclones tropicais (Hall & Jewson, 2007). Adicionalmente, com o estabelecimento de estações de observações do oceano, estações de superfície, e satélites meteorológicos, o volume de dados também está aumentando, resultando em um problema de *big data* (grandes volumes de dados) para prever o movimento de CTs. Inteligência Artificial (IA) é um campo da ciência da computação que se ocupa em desenvolver sistemas que simulam comportamentos inteligentes, especialmente humanos. Aprendizado de Máquina (*Machine Learning*), é um subconjunto de IA que refere-se ao desenvolvimento de sistemas de aprendizagem, ou seja, programas de computador capazes de aprender as regras (ou leis) que descrevem uma tarefa, produzindo modelos matemáticos através dos dados através dos dados observados (Moroney, 2021). Por outro lado, Aprendizado Profundo (*Deep Learning*) é uma classe de algoritmos de Aprendizagem construídos através Redes Neurais Artificiais profundas (Zhang et al., 2023; Prince, 2024). O Aprendizado Profundo poder ser definido, simplesmente, como uma metodologia guiada por dados para produzir modelos que descrevem os sistemas sem muita intervenção humana, isto é, automaticamente extrair e aprender os padrões presentes nos dados que melhor representem o sistema que se pretende modelar (Diesenroth et al., 2020). As redes neurais artificiais modelam as relações lineares e não-lineares entre as variáveis dependentes (*input*) e independentes (*output*) através dos dados observados do problema em causa. O modelo final é representado pela arquitectura e tipologia das definidas pelo usuário, e pelos coeficientes destas redes que são ‘aprendidos’ que são ajustados automaticamente através do treinamento repetitivo e interativo do modelo (Brownlee, 2019; Prince, 2024). Esta abordagem de modelação já tem mostrado resultados satisfatórios na modelação de diversos problemas do mundo real. Os CTs possuem mecanismos dinâmicos complexos e podem ser afectados por vários factores, incluindo o fluxo do vento, efeito beta, condições da estrutura adjacente, temperatura da superfície do mar, e circulação de meso-escala (Chan & Kepert, 2010). Portanto, pela eficiência na modelação de sistemas complexos, redes neurais vêm sendo aplicadas para prever a trajectória de ciclones tropicais, com por exemplo nos estudos feitos por (Wang et al., 2023); (Giffard-Roisin, et al., 2020) e (Boussieux et al., 2022). Estes estudos mostram como as redes neurais podem ser aplicadas para modelar os CTs, e que é necessário desenvolver modelos capazes de incorporar o máximo de informação relevante sobre o

sistema, no caso CTs, para que o mesmos sejam mais generalista (tenha uma boa performance) em situações futuras, não antes vista pelo modelo.

Embora estudos recentes tenham mostrado que redes neurais artificiais podem modelar sistema de tempestade como CTs, poucos estudos têm explorado o uso de tecnologias de Aprendizado de Máquina e Aprendizado Profundo para desenvolver modelos capazes de prever a trajectória de CTs no Sudoeste do oceano Índico Sul. Portanto, este estudo tem como objectivo propor uma abordagem de previsão guiada-por-dados baseada em algoritmos de Aprendizado de Máquina, especialmente Aprendizado Profundo, ou seja, redes neurais profundas para criar modelos que incorporem dados relacionados a trajectória e a informação meteorológica do CT para prever a posição do centro dos mesmos nos próximos instantes ao longo do seu percurso.

1.1. Objectivos

1.1.1. Objectivo Geral

Desenvolver modelos de previsão de trajectórias de Ciclones Tropicais (CTs) de 6-72h sobre a Bacia do Sudoeste do Oceano Índico (SOI) através de tecnologias de aprendizagem profundo.

1.1.2. Objectivos Específicos

- Construir modelos baseados redes neurais artificiais para prever trajectória de CTs;
- Treinar e Avaliar desempenho dos modelos nos dados de trajectória de CTs sobre o SOI de 1993-2023;
- Sugerir um ou mais modelos baseados em redes neurais artificiais para prever trajectórias de CTs sobre o SOI;
- Comparar a performance dos modelos usados neste estudo com os modelos usado actualmente na região.

1.2. Definição do Problema

Os Ciclones Tropicais (CTs) representam alguns dos eventos meteorológicos mais extremos e destrutivos, causando impactos significativos em termos de danos materiais, perdas humanas e perturbações socioeconômicas. A previsão precisa e rápida de suas trajetórias é essencial para a mitigação desses impactos, permitindo a adoção de medidas preventivas mais eficazes. No entanto, prever o deslocamento de um CT continua sendo um grande desafio devido à complexidade e à natureza não linear

das interações entre diversos fatores atmosféricos e oceânicos, como padrões de circulação, gradientes térmicos e interações com sistemas meteorológicos adjacentes.

Embora modelos numéricos tradicionais tenham sido amplamente utilizados para esse fim, eles apresentam limitações, como dependência de condições iniciais, alto custo computacional e dificuldades na captura de variações de pequena escala. Métodos baseados em aprendizado profundo (Deep Learning, DL) têm demonstrado grande potencial na modelagem de fenômenos altamente não lineares, sendo capazes de identificar padrões espaço-temporais complexos em grandes volumes de dados. No entanto, seu uso na previsão de trajetórias de CTs ainda está em estágio de desenvolvimento e requer uma avaliação detalhada de sua eficácia na Bacia do Sudoeste do Oceano Índico (SOI), uma região altamente impactada por esses fenômenos.

Diante desse contexto, este estudo propõe a aplicação de redes neurais profundas para prever as trajetórias de CTs no SOI, explorando sua capacidade de aprendizado autônomo de padrões de deslocamento ao longo do período de 6–72 horas.

1.3. Justificativa

O Sudoeste do Oceano Índico (SOI) é uma região de grande biodiversidade e importância socioeconômica para países costeiros e ilhas que dependem do oceano para subsistência. No entanto, essa região está altamente vulnerável a eventos climáticos extremos, como ciclones tropicais, tempestades e inundações, que ameaçam vidas e economias locais. Diante desse cenário, há uma necessidade urgente de aprimorar os mecanismos de monitoramento e previsão de CTs para mitigar seus impactos. Tecnologias de aprendizado profundo (DL) têm demonstrado eficácia na modelagem de sistemas complexos e podem oferecer previsões mais precisas e rápidas. Este estudo propõe a aplicação dessas técnicas para aprimorar a previsão das trajetórias de CTs no SOI, contribuindo para a segurança e a resiliência das comunidades afetadas. África. No entanto, o bem-estar das pessoas que dependem do oceano para alimentação, rendimento e subsistência, principalmente em países costeiros, é vulnerável a alterações climáticas, ciclones, tempestades e inundações, assim como a actividade humana. Portanto, sabendo da exposição e situação económica dos países e ilhas da região, surge a necessidade de se desenvolver mecanismos monitoria e mitigação dos impactos dos CTs sobre os mesmos. Neste caso, sugere-se neste estudo o uso de tecnologias de aprendizado profundo, que têm demonstrado eficácia em vários outros campos (medicina, agricultura, aviação, etc), para previsão de trajectória de CTs sobre o SOI.

1.4. Estrutura do Trabalho

Este trabalho está dividido em 6 capítulos principais, dos quais o primeiro capítulo apresenta a introdução e os objectivos do trabalho. No Capítulo 2 :, descreve-se de forma resumida a Área de Estudo e apresenta-se a Revisão Bibliográfica onde se destaca alguns conceitos básicos ligados aos ciclones tropicais, assim como os fundamentos do aprendizado de máquinas, importantes para o desenvolvimento desta pesquisa e, também de alguns estudos relacionados a este trabalho. No Capítulo 3 :, Material e Método, explica-se de forma detalhada, o processo de elaboração desta pesquisa e o material necessário para o desenvolvimento da mesma. No Capítulo 4 :, são apresentados e discutidos os resultados obtidos e por conseguinte, são seleccionados os modelos com a melhor performance. No Capítulo 5 :, estão indicadas as considerações finais e as recomendações para trabalhos subsequentes, assim como para a secção de Meteorologia do departamento de Física da UEM e o INAM. Por fim, no sexto capítulo são apresentadas as referências bibliográficas usadas na elaboração deste trabalho.

Capítulo 2 : Revisão Bibliográfica

2.1. Área de Estudo

A região sudoeste do oceano Índico (SOI), limitada a oeste pelo continente Africano, situa-se aproximadamente entre 0° S (equador) e 40° S, de 30° E (costa africana) a 100° E (**Figura 1**). Esta bacia banha várias Ilhas, incluindo: Comores, Mayotte, Madagáscar, Farquhar, Tromelin, Seicheles, La Reunião, Maurício, Agalega, Saint Brandon, Rodrigues e o Arquipélago de Chagos, e Países como: África do Sul, Moçambique, Tanzânia e Quênia (Leroux et al., 2018).

O SOI é a região oceânica com maior biodiversidade de África e a segunda com maior biodiversidade do mundo, no entanto, o bem-estar das pessoas que dependem do oceano para alimentação, rendimento e subsistência é vulnerável a alterações climáticas, ciclones, tempestades e inundações, assim como a actividade humana (UNEP-Nairobi Convention, 2015). Os sistemas tropicais, como Ciclones Tropicais que se formam nesta região são monitorizados pelo Centro Meteorológico Especializado Regional *La Réunion*, gerido pela Météo-France (Leroux et al., 2018).

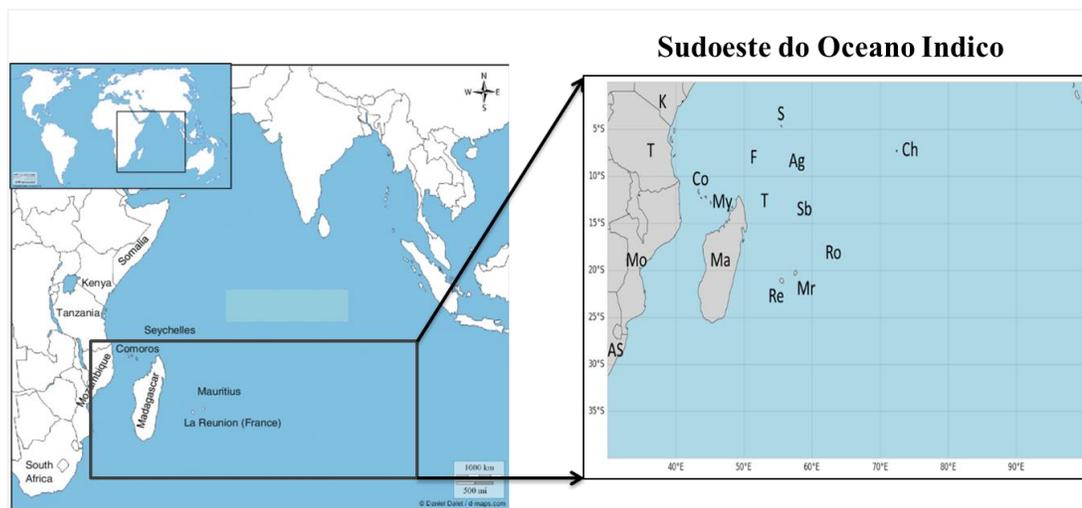


Figura 1: Bacia oceânica do SOI. As letras AS, Mo, T e K indicam África do Sul, Moçambique, Tanzânia e Quênia, respectivamente. Letras Co, My, Ma, F, T, S, Re, Mr, Ag, Sb, Ro, e Ch indicam as ilhas de Comores, Mayotte, Madagáscar, Farquhar, Tromelin, Seicheles, La Reunião, Maurício, Agalega, Saint Brandon, Rodrigues e o Arquipélago de Chagos, respectivamente. Algumas ilhas ou atóis são demasiado pequenos para que as suas fronteiras possam ser vistas no mapa.

2.2. Ciclones Tropicais

Ciclones tropicais são sistemas de tempestade intensos, de escala sinóptica que se formam nos trópicos. Como todos os ciclones, os ciclones tropicais têm um centro de baixa pressão próximo do nível do mar. Além disso, os ventos de baixa altitude giram ciclonicamente (sentido horário no hemisfério sul) em torno da tempestade e em forma de espiral em direção ao centro da tempestade (Stull, 2015).

Os ciclones tropicais (**Figura 2**) são chamados de furacões sobre o Atlântico, Caribe, Golfo do México, e Oceanos do Pacífico oriental. No Pacífico ocidental, Eles são chamados de tufões, e sobre o Oceano Índico e perto da Austrália são chamados de ciclones tropicais (Stull, 2015).

Ciclones tropicais diferem de ciclones de latitude média, pois os ciclones tropicais não estão associados a frentes. Os ciclones tropicais têm núcleos quentes, enquanto os ciclones de latitudes médias têm núcleos frios. Por outro lado, Ciclones tropicais podem persistir duas a três vezes mais do que um ciclone de latitudes médias (Stull, 2015).



Figura 2: Imagem de satélite de Idai às 11h35 UTC de 14 de março de 2019. **Fonte:** (Kolstad, 2021)

2.2.1. Estrutura de Ciclones Tropicais

Os ciclones tropicais são compostos por tempestades. Perto do centro (núcleo) do ciclone tropical existe círculo de tempestades chamado **Parede do olho** (a parte mais violenta da tempestade). O topo dessas tempestades pode se encontrar na parte inferior da estratosfera de 15 a 18 km de altura. A base das tempestades geralmente encontra-se em altitudes baixas: na camada limite planetária. Assim, os ciclones tropicais abrangem a profundidade da troposfera tropical. As bigornas¹ de cada uma das tempestades na parede do olho se fundem em uma grande nuvem quase circular que é visível nas imagens de satélite (**Figura 2**). Essas bigornas podem se espalhar por cerca de 75 a 150 km de distância da parede do olho. Portanto, os diâmetros dos ciclones tropicais são aproximadamente 10 a 20 vezes da sua profundidade. No meio da parede do olho está uma região com tempo mais calmo chamado de olho, com temperaturas quentes, com subsidência de ar e poucas ou nenhuma nuvem. O Diâmetro do olho ao nível do mar é de 20 a 50 km. O olho é cônico, com o diâmetro maior no topo da tempestade (**Figura 3**) (Stull, 2015).

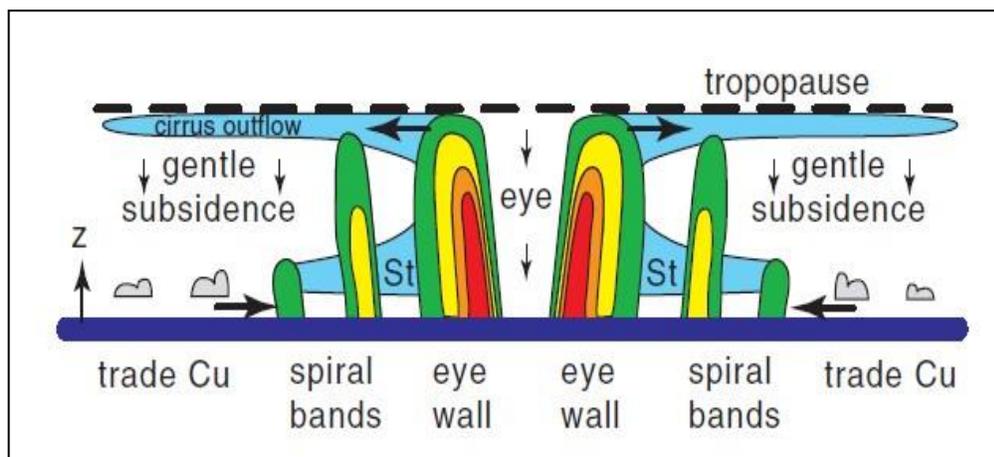


Figura 3: Secção vertical através de um ciclone tropical. As cores verde, amarelo, laranja e vermelho sugerem chuvas moderadas a fortes, como visto por radar. O azul claro representa nuvens estratiformes (St). Altitude da tropopausa ≈ 15 km. Largura do ciclone tropical ≈ 1500 km. Ventos alísios de bom tempo nuvens cumulus (Cu) são esboçadas com sombreamento cinza. Setas; flechas mostram direções do vento. **Fonte:** (Stull, 2015).

¹ É uma nuvem *Cumulonimbus*, com forma de bigorna em seu topo

2.2.2. Intensidade e distribuição geográfica

Pressão mais baixa ao nível do mar, na região do olho e ventos mais rápidos na baixa troposfera indicam ciclones tropicais mais fortes. Várias escalas de ciclones tropicais foram concebidas para classificar a intensidade dos ciclones tropicais, conforme resumido a seguir (Stull, 2015).

- **Escala de Saffir-Simpson**

Na escala de Saffir-Simpson a intensidade do ciclone tropical é definida apenas pela velocidade do vento. A escala varia de categoria 1, para um ciclone tropical fraco até a categoria 5 para um ciclone tropical muito forte (**Tabela 1**) (Stull, 2015). Uma descrição dos danos esperados para cada categoria está na **Tabela 2**.

Tabela 1: Definição da escala de categorias de CTs com base no vento máximo sustentado de 1 minuto, medido na altura padrão do anemômetro de 10 m. Aplica-se no Oceano Atlântico, e no leste do Pacífico desde a costa das Américas até 180 ° W (Stull, 2015).

Categoria	velocidade do vento		
	m.s ⁻¹	km.h ⁻¹	nós
1	33 - 42	119 - 153	64 - 82
2	43 - 49	154 - 177	83 - 95
3	50 - 58	178 - 209	96 - 113
4	59 - 69	210 - 249	114 - 135
5	> 69	> 249	> 135

Tabela 2: Escala de vento do furacão Saffir-Simpson - Descrição dos danos esperados (Stull, 2015)

Categoria	Danos esperados
1	Ventos muito perigosos irão produzir alguns danos.
2	Ventos extremamente perigosos causarão danos extensos.

3	Danos devastadores ocorrerão.
4	Danos catastróficos ocorrerão.
5	Danos catastróficos ocorrerão.

- *Outras escalas de ciclones tropicais*

Escalas adicionais de intensidade de ciclones tropicais (com diferentes definições de velocidade do vento e nomes das categorias) foram definidos por diferentes agências de ciclones tropicais, por exemplo:

- Austrália (Australian Bureau of Meteorology, para Hemisfério Sul, a leste de 90 ° E)
- Índia (*Regional Specialized Meteorological Center*, Nova Delhi, para o Oceano

Índico, Hemisfério Norte, entre 45 ° E e 100 ° E)

- Météo-France (para Hemisfério Sul, a oeste de 90 ° E, **Tabela 3**).

Tabela 3: Escala usada pela Météo-France (para Hemisfério Sul, a oeste de 90 ° E) (Stull, 2015).

Categoria	Velocidade do vento	
	km.h ⁻¹	nós
Distúrbio Tropical	< 50	< 28
Depressão Tropical	51 - 62	28 - 33
Tempestade Tropical Moderada	63 - 88	34 - 47
Tempestade Tropical Severa	89 - 117	48 - 63
Ciclone Tropical	118 - 165	64 - 89
Ciclone Tropical Intenso	166 - 212	90 - 115
Ciclone Tropical muito Intenso	> 212	> 115

A **Figura 1** mostra as regiões de maior frequência de ciclones tropicais e mostra trajetórias típicas de tempestades. Os ciclones tropicais são dirigidos principalmente pela circulação global grande escala. A maioria dos ciclones tropicais se forma entre 10° e 30° de latitude, que é a região dos ventos alísios. Portanto, a maioria dos ciclones tropicais são dirigidos inicialmente de leste para oeste. Uma observação notável na **Figura 4** é que nenhuns ciclones se formam no equador. Além disso, nenhum cruza O equador (Stull, 2015).

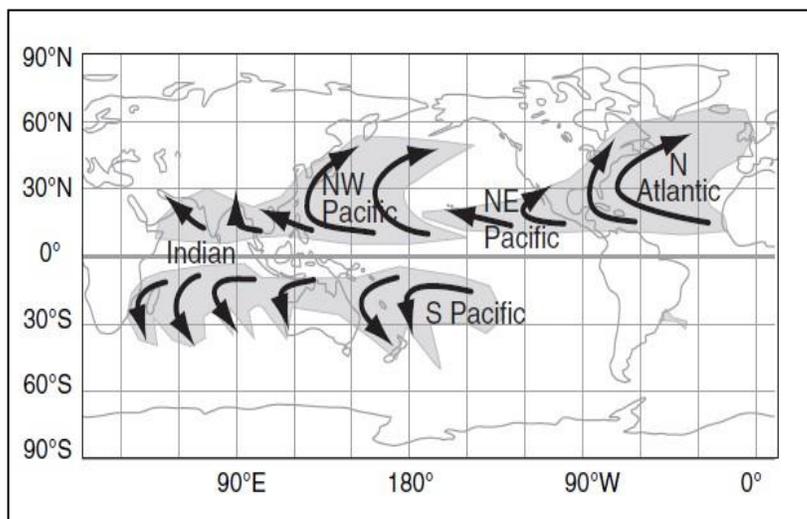


Figura 4: Mapa de localizações de CTs (sombreado cinza), com um quadro geral sentido das suas Trajetórias (setas pretas). **Fonte:** (Stull, 2015).

2.2.3. Requisitos Para a Ciclogénese

Sete condições são necessárias para a formação de ciclones tropicais: uma superfície do mar quente (TSM), Força de Coriolis (FC) diferente de zero, instabilidade condicional não local, alta humidade na troposfera média, fraco cisalhamento do vento no ambiente, aumento de vorticidade em escala sinótica e um gatilho (Stull, 2015). **Tabela 1** dispõe o resumo dos requisitos por cada um dos factores.

Tabela 4: Requisito para *Ciclogénese*

Parâmetro	Descrição do Estado Requerido
-----------	-------------------------------

TSM	A temperatura da superfície do mar (TSM) deve ser aproximadamente 26,5° C ou mais, e as águas superficiais quentes devem ter de pelo menos 50 m de profundidade.
FC	Os ciclones tropicais não podem existir dentro da faixa de cerca de 500 km do equador (ou seja, latitude $\leq 5^\circ$), porque nessas latitudes a força de Coriolis é quase zero (e é exatamente zero no equador).
Instabilidade	Como os ciclones tropicais são feitos de tempestades, o ambiente tropical deve ter instabilidade suficiente para apoiar convecção profunda. Ou seja, deve ter uma camada estável (ou seja, uma tampa) acima de um húmido e quente da camada limite planetária, e a troposfera média deve ser relativamente fria em comparação com a camada limite planetária
Alta humidade na média troposfera	Em uma camada de ar profunda, a humidade deve ser alta em até cerca de 5 km acima do nível do mar. De outra forma, as tempestades em formação não podem continuar a crescer e se organizar em ciclones tropicais
Fracó cisalhamento de vento	Cisalhamento do vento dentro de uma tempestade em formação deve ser fraco para permitir que aglomerados de tempestade se formem. Esses conjuntos de tempestades são os precursores dos ciclones tropicais
Vorticidade em escala sinóptica	Um mínimo relativo de vorticidade relativa na metade inferior da troposfera pode ajudar a organizar as tempestades em um ciclone tropical em formação no hemisfério sul. Caso contrário, quaisquer tempestades que se formassem actuariam de forma independentemente uma das outras

2.2.4. Ciclo da vida dos Ciclones Tropicais

Em locais onde todas as condições necessárias para a formação de um ciclone tropical são observadas, os ciclones tropicais em formação geralmente progridem através dos seguintes estágios de intensificação:

distúrbio tropical, depressão tropical, tempestade tropical e ciclone tropical (Stull, 2015). A **Tabela 5** resume os principais estágios do desenvolvimento do CT

Tabela 5: Ciclo de vida dos Ciclones Tropicais

Sistema	Descrição do Estado de Desenvolvimento
Distúrbio Tropical	um sistema tropical discreto com convecção aparentemente organizada geralmente com 200 a 600 km de diâmetro - com origem nos trópicos ou subtropicais, tendo uma característica migratória não frontal, e mantendo sua identidade por 24 horas ou mais. Este aglomerado de tempestade é visível por satélite como consistindo em tempestades distintas e com regiões de precipitação separadas. Não há nenhum olho e pouca ou nenhuma rotação visível
Depressão Tropical (DT)	A alta pressão no topo da troposfera começa a criar uma circulação térmica, onde o ar é conduzido horizontalmente para fora. Este ar divergente no alto também começa a girar anticiclonicamente. O ar que se move para fora remove as moléculas de ar da região do aglomerado de tempestades, diminuindo assim a pressão da superfície sob o aglomerado de tempestades. Por tanto, ocorre uma "depressão" ao nível do mar, é por isso que esse estágio recebe o nome: depressão tropical.
Tempestade Tropical (TT)	Quando a baixa na superfície é intensa o suficiente para conduzir ventos entre $63 - 88 \text{ km h}^{-1}$ em uma circulação ciclónica fechada, então o sistema é classificado como uma tempestade tropical. Neste ponto, as faixas de chuva das tempestades encontram-se quase completamente envolvidas em um círculo - a futura parede do olho. As bigornas das tempestades geralmente se fundem em uma nuvem central circular densa, que é claramente visível nas imagens de satélite como uma nuvem alta e fria de grande diâmetro.
Ciclone Tropical (CT)	Neste estágio há um olho bem definido cercado por uma parede do olho. As velocidades máximas do vento são maiores que 188 km h^{-1} em uma circulação fechada ao redor do olho. Há uma forte precipitação na parede do olho e o núcleo da tempestade é significativamente mais quente do que o ambiente ao redor. A diminuição da pressão ao nível do mar no olho é significativa, e um aumento do

nível do mar em uma tempestade é possível. A organização da tempestade pode persistir por dias a semanas enquanto se move em bacias oceânicas **Figura 2**

Os ciclones tropicais podem durar semanas porque eles têm a capacidade de criar seu próprio suprimento de combustível de ar quente e húmido a partir da camada limite planetária. Eles criam este combustível, aproveitando o calor e a água do oceano. Ciclones tropicais dissipam-se (ciclólise tropical) quando eles se mudam para um local onde eles não podem criar seu próprio combustível, ou se forem destruídos por sistemas climáticos de grande escala (Stull, 2015).

2.2.5. Climatologia das Trajectórias de Sistemas Tropicais sobre o SOI

Os sistemas tropicais estão distribuídos uniformemente ao longo da bacia do SOI de 5° a 35° S, conforme mostra-se na **Figura 5**. Estes sistemas movem-se principalmente para sudoeste, de acordo com o ambiente predominante e direção ambiental (ventos alísios de leste) e com o efeito beta, antes de se recurvarem para sul ou sudeste para sua transição extratropical (Leroux et al., 2018). Sobre o Oceano Índico aberto, CTs que são equivalentes, em intensidade, a Furacões ou tufões que estão principalmente confinados na faixa de latitude de 10°- 25° S.

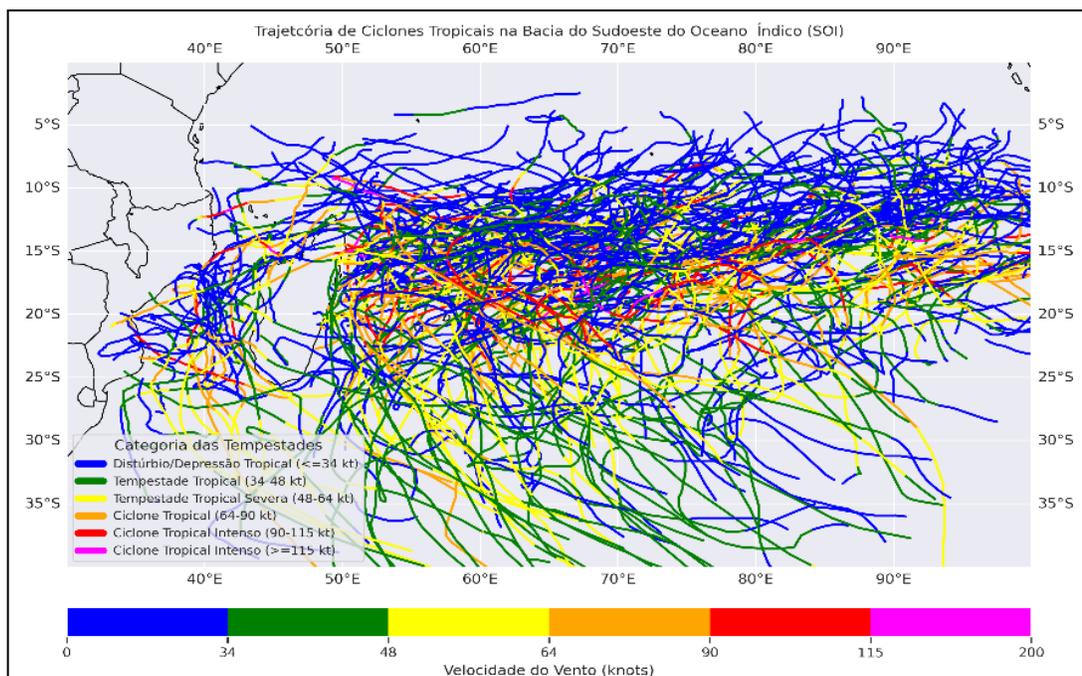


Figura 5: Trajetórias dos sistemas tropicais sobre o SOI com legenda de intensidade baseado na escala da **Tabela 3** dos dados IBTrACKs durante o período 1993–2023.

No Canal de Moçambique, no entanto, a actividade muda aproximadamente 5° para o sul e os CTs ocorrem com mais frequência entre 15° – 30° S. Segundo (Leroux et al., 2018), esta mudança se deve, principalmente, a uma posição mais distante do cavado das monções do Pólo (que gera a maioria das tempestades SOI) e a TSM ser mais altas nessas latitudes em relação a TSM em oceano aberto.

2.2.6. Modelos Previsão de Trajetórias dos CTs

A tarefa de previsão abrange a trajetória, intensidade, tamanho, estrutura dos CTs e tempestades, chuvas e tornados associados. No entanto, a maioria dos modelos de previsão concentra-se na produção de previsões de trajetória e de intensidade em um determinado intervalo de tempo. Os modelos de previsões operacionais actuais de TCs podem ser classificadas em **Modelos Dinâmicos**, **Modelos Estatísticos** e **Modelos Estatístico-Dinâmicos** (Cangialosi, 2020).

Os modelos dinâmicos, também conhecidos como modelos numéricos, utilizam supercomputadores poderosos para simular a evolução dos campos atmosféricos usando métodos dinâmicos e equações termodinâmicas (ECWMF, 2019). Os Modelos estatísticos aproximam relações históricas entre o comportamento da tempestade e as características específicas da tempestade e, em geral, não consideram explicitamente o processo físico (Aberson, 1998). Os modelos estatístico-dinâmicos usam técnicas estatísticas, mas incluem ainda variáveis atmosféricas fornecidas por modelos dinâmicos (DeMaria et al., 2005). Por último, os modelos de conjunto (*ensemble*) combinam as previsões feitas por múltiplas execuções de um ou vários modelos (Cangialosi, 2020). Além disso, os modelos de consenso normalmente combinam previsões operacionais através de média simples ou ponderada (Simon, et al., 2018).

Os desenvolvimentos recentes na estatística e computação permitem que modelos de aprendizado de máquina, especialmente redes neurais profundas, empreguem múltiplas técnicas de processamento de dados para processar e combinar informações de uma ampla gama de fontes e criar arquiteturas sofisticadas para modelar relações espaço-temporais das trajetórias CTs (Leroux et al., 2018).

2.3. Fundamentos de Aprendizado de Máquinas

2.3.1. Descrição de Algoritmos de Aprendizado de Máquinas

Tradicionalmente, programação envolve escrever uma serie de regras (algoritmo), expressas em uma linguagem de programação, que actuam sobre os dados e que geram respostas (ou executam quaisquer tarefas). Por outro lado, algoritmos de aprendizagem de máquina referem-se ao tipo de algoritmos, ou seja, programas de computador, capazes de aprender as regras (ou leis) que descrevem uma tarefa através dos dados (Moroney, 2021). A **Figura 6** mostra esquematicamente como estas abordagens actuam sobre os dados. Enquanto na programação tradicional (**Figura 6 (a)**) as regras que geram certa resposta através dos dados são previamente definidas, no aprendizagem de máquinas (**Figura 6 (b)**) os programas são configurados para descobrir as regras (ou relações) que associam os dados às respostas.

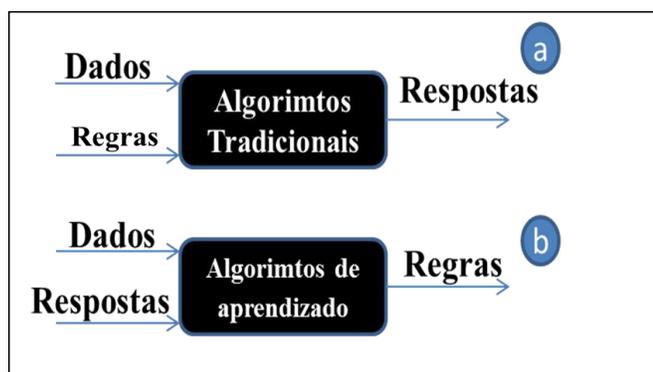


Figura 6: Programação Tradicional (a) e Aprendizado de Máquinas (b). **Fonte:** Adaptado de (Moroney, 2021)

Formalmente, diz-se que um programa está a aprender da Experiência **E** com respeito a uma classe de tarefa **T** e medida de performance **P**, se a sua performance na tarefa **T** conforme medida por **P**, melhora com **E** (Goodfellow et al., 2016). Onde:

❖ Tarefa, **T**

Tarefas de *Machine Learning* (ML) são geralmente descritas em termos de como o sistema deve processar um exemplo (amostra, ponto), e não a aprendizagem em si. Um exemplo é uma coleção de atributos (variáveis, recursos) que foram quantitativamente medidas de algum objecto ou evento que se pretende que o sistema de ML processe. Tipicamente, os exemplos são representados como um vector $x \in \mathbb{R}^n$ onde cada elemento x_i do vector é um outro atributo (Goodfellow et al., 2016).

Existem vários tipos de tarefas que podem ser resolvidas usando ML, os exemplos mais comuns incluem Classificação e Regressão. Na **classificação** deseja-se que sistema especifique quais das k categorias um *input* (exemplos, entrada) pertence. Para resolver esta tarefa, o algoritmo geralmente produz uma função $f : \mathbb{R}^n \rightarrow \{1, \dots, k\}$. Quando $y = f(x)$, então o modelo designa um exemplo descrito pelo vector x a uma categoria identifica por um código numérico y . **Regressão** é o tipo de tarefa na qual o programa estima (ou prevê) um valor numérico real dado algum *input*. Para resolver esta tarefa, o algoritmo geralmente produz uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Este tipo de tarefa é similar a classificação, excepto que o *output* é diferente (Goodfellow et al., 2016).

❖ Medida de Performance, P

De modo avaliar a habilidade predicativa do algoritmo de ML, deve-se desenvolver uma medida quantitativa de sua performance. Geralmente, esta medida de performance **P** é específica para a tarefa T sendo executada pelo sistema. Para Regressão, a medida comum é o Erro Médio Quadrado (Goodfellow et al., 2016)

Geralmente o interesse está na performance do algoritmo de ML nos dados que este não ‘observou’ antes no treinamento, já que deste jeito se determina como será a performance do programa quando esta for aplicada no mundo real. Portanto, a avaliação da performance do programa é feita usando o conjunto de dados designado por conjunto de teste, que é uma parte isolada dos dados usados para o treinamento do sistema de ML (Goodfellow et al., 2016).

❖ Experiência, E

Algoritmos de ML, em geral, podem ser categorizados como **Supervisionado** ou **Não Supervisionado** pelo tipo experiência que lhes são permitidas de ter durante o processo de aprendizagem. A maioria dos algoritmos de aprendizagem podem se entendidos pela forma que são permitidos a experimentar um conjunto de dados. Um conjunto de dados é uma coleção de vários exemplos, também chamados de amostras ou pontos. Neste contexto, um algoritmo de aprendizagem **não supervisionada** experimenta um conjunto de dados contendo vários atributos (x), e de seguida aprende as propriedades uteis da estrutura deste conjunto de dados. A forma mais comum desta aprendizagem é a Aglomeração (*Clustering*). Por outro lado, um algoritmo de aprendizagem **supervisionado** experimenta um conjunto

de dados contendo vários vectores de atributos x (*input*), mas cada um destes vectores é associada a um alvo (*output*), ou seja, variável dependente y (Goodfellow et al., 2016).

Rigorosamente falando, aprendizagem **Não Supervisionada** envolve observar vários exemplos de um vector aleatório x e, implicitamente ou explicitamente, aprender a distribuição de probabilidade $p(x)$, ou alguma propriedade relevante da distribuição, enquanto aprendizagem **Supervisionada** envolve observar vários exemplos de um vector aleatório x e associar a um valor ou vector y , e aprender a prever y a partir de x , geralmente estimando $p(y|x)$ (Goodfellow et al., 2016).

2.3.2. Regressão Linear como um exemplo de Aprendizagem de Máquina

Para ser mais concreto e introduzir alguns conceitos relacionados a *machine learning*, esta secção descreve como ML é aplicada para resolver tarefa de Regressão Linear.

Regressão linear resolve um problema de regressão. Em outras palavras, o objetivo é construir um sistema que possa levar um vector $x \in \mathbb{R}^n$ como *input* e estima valor de um escalar $y \in \mathbb{R}$. No caso de regressão linear, o *output* é uma função linear do *input*. sendo y' o valor que o modelo prevê que o y deva tomar. O *output* é definido como:

$$y' = w^T \cdot x \quad (2.1)$$

Onde $w \in \mathbb{R}$ é um vector de parâmetros (os pesos) que determinam como cada atributo afeta a previsão. Os parâmetros são valores que controlam o comportamento do sistema ou modelo (Goodfellow et al., 2016). Neste caso, a tarefa **T** é prever y dos x produzindo $y' = w^T \cdot x$.

Para medir a performance do modelo pode-se calcular o error médio quadrado (*mean squared error*, MSE) dos dados de teste. Se y'^{teste} fornece a previsão do modelo no conjunto de teste, então o erro médio quadrado é dado por:

$$MSE_{teste} = \frac{1}{m} \sum (y'^{teste} - y^{teste})^2 \quad (2.2)$$

Este decresce quando $y'^{teste} = y^{teste}$. Por outro lado, este erro pode ser expresso em termos da distancia euclidiana:

$$MSE_{teste} = \frac{1}{m} \|y'^{teste} - y^{teste}\|_2^2 \quad (2.3)$$

Portanto o erro aumenta quando a distancia entre o valor previsto e real aumenta (Goodfellow et al., 2016).

Para fazer um algoritmo de ML, é necessário criar um algoritmo que irá melhorar os valores os valores w_i na equação [1.1] de tal modo que reduza o erro de MSE_{teste} quando é permitido que o algoritmo ganhe algum conhecimento observando o conjunto de treinamento (x^{trein}, y^{trein}) . Uma forma intuitiva de assim o fazer, é apenas minimizando o MSE no conjunto de treinamento (Goodfellow et al., 2016). Para algoritmos simples como Regressão linear, existem dois principais métodos para achar os w_i e minimizar o MSE_{trein} :

- **Forma Direta**

É um procedimento para calcular, de forma direta, os valores de w_i que minimizem os valores MSE_{trein} . A expressão matemática que fornece os w_i diretamente é:

$$w = (x^{treinT} \cdot x^{trein})^{-1} \cdot x^{treinT} \cdot y^{trein} \quad (2.4)$$

- **Método de Optimização Iterativa**

A ideia central consiste em gradualmente ajustar os valores de w_i para minimizar os valores MSE_{trein} até que o algoritmo convirja a um conjunto de w_i que representem um mínimo global de MSE_{trein} . Tal algoritmo de optimização é designado por Gradiente Descente (*Gradiente Descent, GD*) (também descrito nos Anexos). Matematicamente, este é expresso como:

$$w_i = w_i - \alpha \frac{\partial MSE_{trein}(w)}{\partial w_i} \quad (2.5)$$

Onde α é designado por taxa de aprendizagem. De modo geral, este algoritmo repetitivamente mede o gradiente local do error MSE_{trein} com respeito a cada um dos parâmetros w_i do modelo e move-se na direcção mais íngreme da diminuição do MSE_{trein} (Goodfellow et al., 2016).

O termo regressão linear é frequentemente usado para se referir a uma função ligeiramente sofisticada do que a função (2.1), geralmente este apresenta mais um parâmetro, o termo de intercepção b :

$$y' = \mathbf{w}^T \cdot \mathbf{x} + b \quad (2.6)$$

b é chamado de viés (*bias*). Esta terminologia vem do ponto de vista de que o *output* da transformação dos atributos é enviesado (*biased*) para b (Goodfellow et al., 2016).

2.3.2.1. Capacidade, *Overfitting* e *Underfitting*

O desafio central em ML é de treinar o modelo de tal modo que este tenha uma boa performance em novos, anteriormente não visto, *inputs*, não apenas naqueles nos quais o modelo foi treinado. A habilidade reslizar boas previsões nos dados não observados anteriormente (no treino) é chamado de **Generalização**. O erro do modelo medido por MSE_{trein} é chamado de erro de treinamento, e reduz-se este erro durante o treinamento. Até agora, o processo de aprendizagem foi descrito como um problema optimização, o que separa ML de optimização é que se deseja que erro de generalização, chamado de erro de teste, seja mínimo também (Goodfellow et al., 2016).

Tipicamente, o erro de generalização de um sistema de ML é medido medindo a sua performance no conjunto de exemplos de teste que foram separados dos dados de treinamento. No exemplo do modelo de regressão, treina-se o modelo minimizando o erro de treinamento $\frac{1}{m} \|X^{\text{trein}} \cdot \mathbf{w} - y^{\text{trein}}\|_2^2$, no entanto o que mais importa é a redução do erro de teste $\frac{1}{m} \|X^{\text{teste}} \cdot \mathbf{w} - y^{\text{teste}}\|_2^2$ (Goodfellow et al., 2016). Portanto, a redução do erro de treinamento é uma forma de atingir um modelo que generalize para casos futuros do mundo real, e o seu potencial é avaliado no teste.

Os factores que determinam como o modelo vai se comportar em situação futuras são:

- Capacidade de minimizar o Erro de treinamento;
- Capacidade de reduzir a lacuna entre erro de treinamento e teste.

Estes dois factores correspondem aos 2 desafios centrais do ML: *overfitting* (*super-ajuste*) e *Underfitting* (*sobre-ajuste*). ***Underfitting*** ocorre quando o modelo não é capaz de reduzir o valor do erro no conjunto

de treinamento suficientemente, e *overfitting* ocorre quando há uma lacuna significativa entre o erro de teste e treinamento (Goodfellow et al., 2016). Controla-se o modelo mais provável de sofrer de *Underfitting* ou *overfitting* alterando a capacidade do modelo. Informalmente, a capacidade de um modelo é definida pela sua capacidade de se adaptar (*fit*) a uma ampla variedade de funções através da (complexidade) sua estrutura. Modelos com baixa capacidade têm dificuldades para se adaptar aos dados de treino (*underfit*). Modelos com alta capacidade podem se adaptar perfeitamente aos dados de treino memorizando padrões dos dados de treinamento que não servem muito para os dados de teste. De modo geral, os dois casos geram péssimos modelos, o desejável é situar a configuração dos parâmetros em um ponto intermediário dos dois extremos (Goodfellow et al., 2016).

2.3.2.2. Regularização

É possível controlar a performance do algoritmo de aprendizagem escolhendo que tipo de função se pretende que o algoritmo escolha do seu espaço de hipotético de soluções, assim também como, controlando a quantidade destas funções. Também pode se indicar a preferência de soluções que se pretende que o algoritmo desenvolva. Para tal, pode se alterar o critério de treinamento para regressão linear para incluir o decaimento de pesos (*weight decay*). Para realizar regressão linear com *weight decay*, minimiza-se uma soma compreendendo o MSE_{trein} e a normal dos parâmetros w em cada actualização. Este critério (que expressa a preferência de solução) é representado por $J(w)$. E para expressar preferência por soluções que tenham valores reduzidos da normal L^2 *norm* dos w é :

$$J(w) = MSE_{\text{trein}} + \lambda \cdot w^T \cdot w^T \quad (2.7)$$

Onde λ é uma valor escolhido de antemão que controla a força da preferência de solução imposta sobre os valores de w , e expressa tendência de *overfit* ou *Underfit* (Goodfellow et al., 2016).

De modo geral, pode regularizar um modelo que aprende uma função $f(x, \theta)$ adicionando uma penalização chamada de regularizador. No caso de *weight decay*, a penalização é $\Omega(w) = w^T \cdot w$, cuja força é determinada por λ . Na pratica existem vários outros regularizadores, e todas estas abordagens são designadas por regularização, e formalmente é definida como sendo qualquer mudança feita sobre o algoritmo que destinada a reduzir o seu erro de generalização, mas não o seu erro de treinamento (Goodfellow et al., 2016).

2.3.2.3. Hiperparâmetros e conjunto de validação

Muitos algoritmos (dos simples aos complexos) de ML têm várias definições (*settings*) que podem ser usados para controlar o comportamento do algoritmo de aprendizagem (introduzindo regularização na maioria das vezes). Estas definições são chamadas de hiperparâmetros (*hyperparameters*), os valores deste, diferentemente dos parâmetros w , não adaptados pelo algoritmo de aprendizagem, pelo contrário, são definidos (e ajustados) de antemão pelo usuário (Goodfellow et al., 2016).

Se estes hiperparâmetros forem otimizados nos dados de treinamento, tais hiperparâmetros sempre escolheriam a capacidade máxima do modelo, resultando em *overfitting* dos dados de treinamento. Por outro lado, os dados de teste servem apenas para avaliar a generalização final do modelo, e não devem de alguma forma serem usadas para fazer qualquer escolha sobre o modelo incluindo os hiperparâmetros. Portanto, para evitar estes problemas, é necessário ter um conjunto de exemplos de validação que o algoritmo de treino não tenha visto no seu treinamento. Especificamente, os dados de treino são separados em dois conjuntos desconexos, um para achar os parâmetros w do modelo, o outro para estimar o erro de generalização durante o treino, permitindo otimizar os hiperparâmetros de acordo resposta da performance (Goodfellow et al., 2016).

2.3.3. Aprendizado Profundo

Na secção [2.3.1.] foram introduzidos alguns dos princípios básicos do aprendizado de máquinas, nesta secção são apresentados alguns conceitos fundamentais do aprendizado profundo (Deep learning, DL), assim também como algumas arquiteturas de redes neurais artificiais fundamentais do DL, assim como aplicadas no presente trabalho.

2.3.3.1. Definição do Aprendizado Profundo

Aprendizado Profundo é uma classe de algoritmos de Aprendizado de Máquina que é construída sobre Redes Neurais Artificiais, formalmente se refere ao conjunto de algoritmos e modelos computacionais que são compostos por múltiplas camadas de processamento. Este também tenta aprender representações dos dados com múltiplos níveis de abstração (representação hierárquica dos dados) (Zhang et al., 2023).

2.3.3.2. Componentes de sistemas de DL

Modelos baseados em DL são principalmente compostos por: **Parâmetros, Camadas, Função de activação, hiperparâmetros e método de Optimização.** Onde:

- **Parâmetros**

Conforme dito na secção [2.3.2.3] parâmetros w de um modelo de aprendizagem de maquinas são valores escalares que relacionam os valores de x_i a previsão de y . Do mesmo modo, parâmetros de redes neurais estabelecem a magnitude da conexão entre os diferentes neurónios (ou unidades) das camadas da rede que se pretende otimizar.

- **Camadas**

No DL, camadas de neurónios representam uma coleção de neurónios neurais artificiais que compartilham dos mesmos *inputs*, ou seja, a forma como os neurónios são organizados na rede. Existe três tipos gerais de camadas: camada de entrada (*inputs layer*), camada escondida (*hidden layer*), camada de saída (*output layer*), dependendo da posição na rede (Géron, 2019).

- **Função de activação**

Funções de activação são usados para propagar o *output* de um neurónio de uma camada para a próxima camada (incluindo o *output layer*), determinando a quantidade do *output* deste neurónio que é deixada passar adiante. Existem vários tipos de ativações para diferentes fins, os mais comuns são: *Rectifier Linear Units (ReLU)*, sigmoide e Tangente Hiperbólica (Géron, 2019).

- **Função de perda**

A função de perda (*loss function*) quantifica quão próximo uma rede está dos parâmetros ideais para os quais esta está sendo treinada. Esta função representa o erro de treinamento tal como introduzido na secção [2.3.2.] No entanto, para redes neurais, esta função é expressa em termos do *bias* e *weights*. O erro médio quadrado (uma das funções mais populares) pode ser escrito matematicamente como:

$$L(\mathbf{w}, \mathbf{b}) = \frac{1}{N} \sum_i^N (\mathbf{y}'_i - \mathbf{y}_i)^2 \quad (2.8)$$

- **Hiperparâmetros**

Conforme introduzido na seção Hiperparâmetros e conjunto de validação, estes são parâmetros que são ajustados pelo usuário durante o processo de treinamento para controlar a otimização dos parâmetros do modelo e, a sua seleção garante que o modelo não sofra de *underfitting* nem *overfitting*. Alguns dos hiperparâmetros mais importantes em redes neurais são: taxa de aprendizagem (*learning rate*), regularização, número de neurónios e camadas.

- **Métodos de Optimização**

Conforme introduzido na seção [2.3.2.] pela equação [2.5.], é o conjunto de procedimentos usados para actualizar os parâmetros do modelo (rede) iterativamente até que atingir valores mínimos da função de custo [2.8]. Na seção [2.3.2.] foi introduzido o algoritmo gradiente descendente, mas na prática, este serve de base para o funcionamento de métodos mais sofisticados como o Adam e *Back Propagation* (Géron, 2019).

2.3.4. Redes fundamentais de DL

2.3.4.1. Redes Feed forward

- **Neurónio Biológicos e Neurónios Artificiais**

O neurónio biológico (**Figura 7 (b)**) é uma célula nervosa que fornece a unidade funcional e fundamental para o Sistema nervoso de todos os animais. Os neurónios existem para se comunicarem entre si e transmitirem impulsos eletroquímicos através das sinapses, de uma célula para outra, desde que o impulso seja forte o suficiente para ativar a liberação de substâncias químicas através de uma fenda sináptica. A força do impulso deve ultrapassar um limite mínimo ou as substâncias não serão liberados. O neurónio é composto por uma célula nervosa que consiste em uma **soma** (corpo celular) que possui muitos **dendritos**, mas apenas um **axónio** (Pettersson & Gibson, 2017).

Os neurónios artificiais (**Figura 7 (a)**) recebem informações que, com base nos pesos das conexões, podem ser ignoradas ou passadas para a função de activação (α). A função de activação também tem

capacidade de filtrar dados que recebem determinando à que extensão estes serão passados para próxima camada. Neurónios artificiais são compostos por pesos das conexões, *bias* (permitem que a aprendizagem aconteça dando a rede activação no caso de fraco sinal de entrada) e função de activação. Os *inputs* são os dados a partir dos quais se deseja produzir informações, e os pesos e *bias* das conexões são as grandezas que governam a actividade do neurónio. Tal como acontece na regressão linear [Regressão Linear como um exemplo de Aprendizagem de Máquina, aplica-se um algoritmo de optimização que altera os pesos e *bias* do neurónio durante a fase do treinamento. Matematicamente, um neurónio artificial é representado como:

$$\propto (b + \sum w_i * x_i) \quad (2.9)$$

Assim como os neurónios biológicos não transmitem todos os impulsos eletroquímicos que recebem, os neurónios artificiais são projetados para serem igualmente seletivos. Eles filtram os dados que recebem e agregam, convertem e transmitem apenas determinadas quantidade de informações para o(s) próximo(s) neurónio(s) da rede. À medida que esses filtros funcionam sobre os dados, eles convertem os dados brutos de *input* em informações uteis para o contexto da rede neural de múltiplas camadas (Petterson & Gibson, 2017).

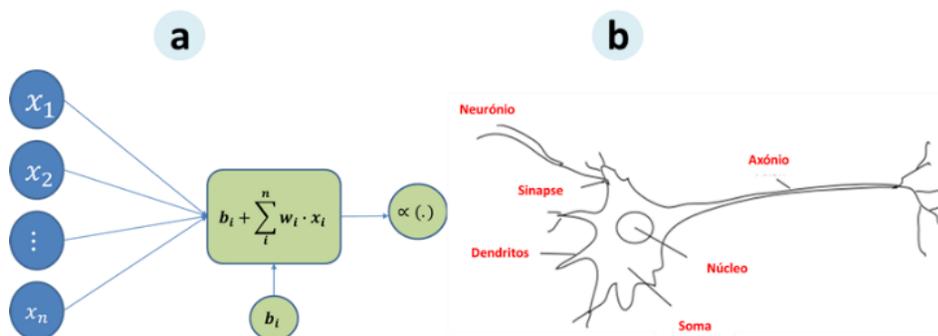


Figura 7: Componentes de: Neurónio Artificial (a) e Neurónio Biológico (b).

Uma rede *Feed forward*, ou *Multi-Layer Perceptron*, aproxima certa função $F(x)$ usando os exemplos fornecidos de uma certa distribuição $P(x)$. Nesse caso, espera-se que a rede *Feed forward* encontre $F(x|w)$ tal que se aproxime da função ideal $F(x)$ o máximo possível durante o treinamento. Os neurónios de uma rede *Feed forward* são organizados em camadas, onde todos neurónios de uma camada

são conectados a todos os neurónios da camada anterior e posterior. A generalização da operação de um neurónio $\alpha (\mathbf{b} + \sum \mathbf{w}_i * \mathbf{x}_i)$ (2.9) para uma camada arbitrária, assumindo que a $k - \text{ésima}$ camada tem N^k Neurónios e o *output* da camada pode ser representado como \mathbf{h}^k , com $h_i^{(k)}$ denotando o i -ésimo elemento do vector do *output*, o *forward pass* poder ser feita através de:

$$\mathbf{h}_j^{(k)} = \alpha (\mathbf{b}_j^{k-1} + \sum_{i=1}^{N^{(k)}} \mathbf{W}_{ji}^{(k-1)} \mathbf{h}_i^{(k-1)}) \quad (2.10)$$

Onde $\mathbf{W}_{ji}^{(k-1)}$ representa matriz dos pesos correspondente a conexão entre $\mathbf{h}_i^{(k-1)}$ e $\mathbf{h}_j^{(k)}$, e \mathbf{b}_j^{k-1} é o bias para calcular $\mathbf{h}_j^{(k)}$. De modo geral, esta equação pode ser representada na forma vectorial como:

$$\mathbf{h}_j^{(k+1)} = \alpha (\mathbf{b}^k + \mathbf{W}^{(k)} \mathbf{h}^k) \quad (2.11)$$

Onde $\mathbf{W}^{(k)} \in \mathbb{R}^{N^{(k)} \times N^{(k-1)}}$ e contém todos os coeficientes e o seu $j, i - \text{ésimo}$ elemento é $\mathbf{W}_{ji}^{(k)}$ da equação [2.11] e $\mathbf{b}^{(k)}$ consiste de todos os *bias* (Zhang et al., 2023). Esquemáticamente, uma rede *feed forward* pode ser representada como ilustrado na **Figura 8**.

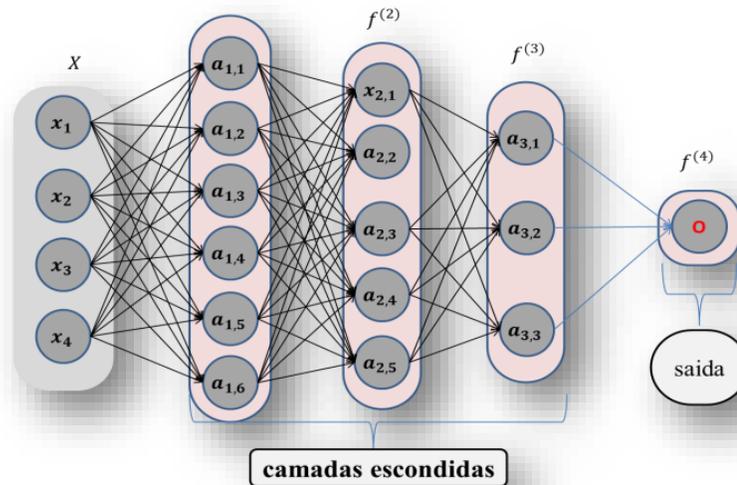


Figura 8: Rede Neural Feed Forward Totalmente Conectada.

A **Figura 8** tem quatro funções $f^{(1)}, f^{(2)}, f^{(3)}, f^{(4)}$ que representam as camadas escondidas [2.10] conectadas em cadeia e $a_{j,i}$ representa a activação de cada um dos $j, i - \text{ésimos}$ neurónios da camada.

Conforme dito anteriormente, as funções de activação decidem à que extensão o sinal de entrada passa para próxima camada. Por outro lado, a função de activação, também, introduz não linearidade à rede neural tornando-as capazes de aproximar funções não lineares (Zhang et al., 2023).

2.3.4.2. Redes Neurais Convolucionais

Rede neural convolucional (*convolutional neural network*, CNN) é um tipo de rede neural *Feed Forward* que aprende recursos em dados estruturais por si só por meio de optimização de filtros (ou *kernels*). Este tipo de rede de aprendizagem profunda tem sido aplicado para processar e fazer previsões a partir de muitos tipos diferentes de dados, incluindo texto, imagens e áudio (Géron, 2019). Embora essas redes sejam aplicadas em varias tipos de dados, a descrição destas será feita considerando a sua aplicação em imagens, que é a aplicação mais comum e intuitiva.

CNN são capazes de extrair padrões e recursos de dados brutos que são determinantes para a realização de uma dada tarefa com sucesso durante a aprendizagem. Por essa razão as CNNs são consideradas extratores de padrões ou características, onde: camadas mais próximas do *input* extraem recursos de baixo nível dos dados brutos, e camadas posteriores, constroem padrões hierárquicos por ‘cima’ informações vindas destas camadas e camadas mais externa (*output layer*) usam estes padrões para prever valores contínuos, em regressão, ou classes em classificação (Raschaka & Mirijalili, 2019). Conforme ilustrado na **Figura 9** CNNs produzem **mapas de características** das imagens de entrada, onde cada elemento deste mapa vem de uma parcela local (**campo recetivo local**) de *pixels* da imagem de entrada.

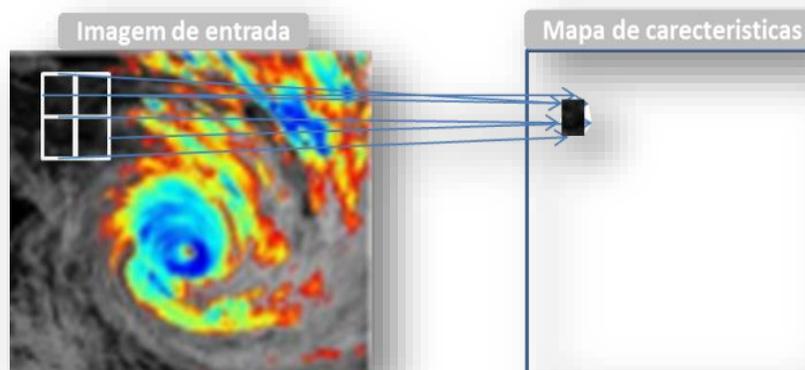


Figura 9: Processamento convolucional de imagem de um ciclone e Construção de Mapas de Características.

As CNNs são ótimas para processar imagens devido a duas principais propriedades:

- ***Sparse Connectivity***: um único elemento no mapa de características está conectado à uma pequena parcela de *pixels* da imagem de entrada.
- ***Parameter-Sharing***: os mesmos coeficientes são usados por diferentes parcelas da imagem de entrada.

Como uma consequência direta destas propriedades, as CNNs são mais eficientes para processar imagens em relação a redes convencionais como redes *Feed Forward (simples)*. No entanto, estas redes são similares a redes neurais *Feed Forward* em vários aspetos, como por os neurónios que têm pesos e *bias* treináveis, e as camadas são hierárquicas e sequenciais (Raschaka & Mirjalili, 2019). CNNs são compostos: **camadas convulsionais** que usam *kernels* para extrair e padrões das imagens, **camadas *pooling*** que resume os *outputs* das camadas de convolução em padrões mais dominantes, e **camada de saída** (output layer) responsável por realizar a previsão (Géron, 2019).

A **Figura 10** ilustra a operação convolucional realizada pela camada sobre uma imagem, onde o *filter* é pequena matriz que contém os pesos de conexão de cada neurónio e *ReLU* é a função de activação aplicada sobre a soma do *bias* e o mapas de característica (produto entre o *kernel* e uma parcela (*patch*) da imagem) que produz o output.

Uma cada Convolutional com dois filtros

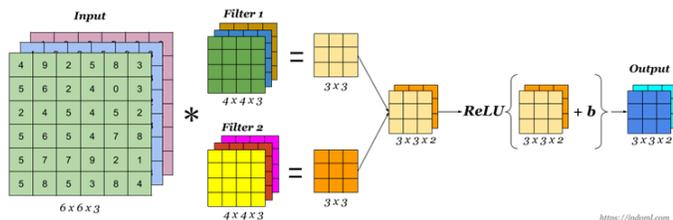


Figura 10: Camada convolucional. **Fonte:** Adaptado de (Zhang et al., 2023)

Os *filters* (ou *kernels*) não são definidos manualmente, pelo contrário, estes são otimizados durante treinamento. As camadas convulsionais podem ser compostos por vários *kernels*, onde cada deles é responsável por gerar um mapa de características ou, detectar a presença de um padrão na imagem.

As camadas de *pooling* são compostas por operações de *downsampling* que são responsáveis pela redução da dimensão espacial das imagens. A **Figura 11** ilustra esquematicamente duas operações de pooling mais comuns em CNNs: **max pooling** é a operação de pooling que encontra o valor máximo para cada parcela (*patch*) do mapa de característica (*feature map*) e retorna o mapa de característica em dimensão reduzida, e o **average pooling** é a operação de pooling que retorna a média dos valores dos *pixels* em cada parcela (*patch*) do mapa de característica (*feature map*) e retorna o mapa de característica em tamanho reduzido (Géron, 2019).

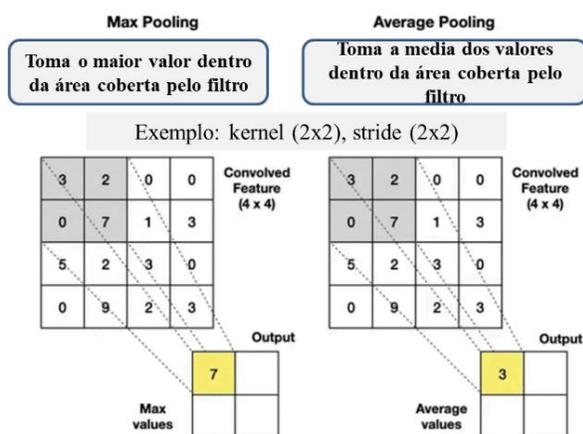


Figura 11: operações de pooling: esquerda - max pooling, direita - average pooling. **Fonte:** adaptado de (Zhang et al., 2023)

A estrutura geral de uma CNN, mostrado na **Figura 12** pode ser dividida em dois componentes: **componente de extração** de padrões (camadas convolucionais e *pooling*) e a **componente de Regressão** (ou classificação, uma rede *Feed Forward*) (Zhang et al., 2023). Embora seja apresentadas apenas duas camadas convolucionais, na prática são empilhadas várias camadas em redes neurais profundas.

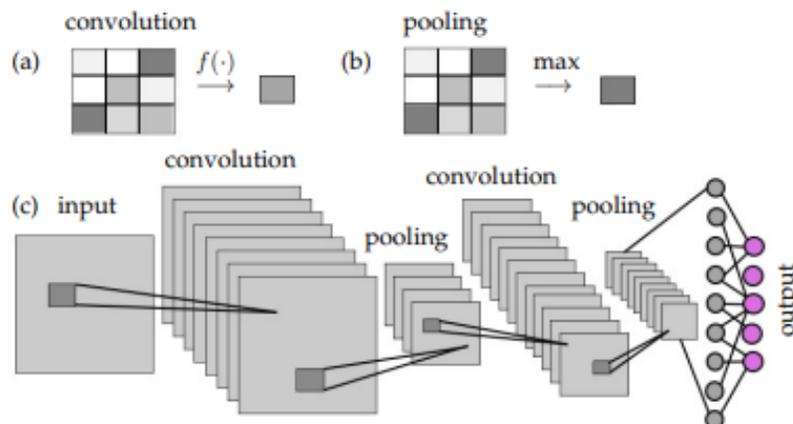


Figura 12: Estrutura geral de uma CNN que inclui as camadas de convolução, *pooling* e *output*. Os quadrinhos cinzento-escuro são os *kernels* de convolução. Para cada camada, vários *kernels* podem ser usados para produzir diferentes mapas de recursos (ou características). **Fonte:** (Brunton & Kutz, 2021)

2.3.4.3. Redes Neurais Recursivas

As Redes Neurais Recursivas (*Recurrent Neural Networks, RNNs*) são introduzidas para resolver problemas onde os exemplos do conjunto de dados podem ser representados como uma sequência de valores. As RNNs aplicam a mesma função recursivamente à cada um dos elementos da sequência, um por um, conservando a relação temporal entre cada um dos elementos (Zhang, Lipton, Li, & Smola, 2023).

a. Estrutura tradicional de Redes Neurais Recursivas

Uma sequência de p elementos pode ser representada como x^1, x^2, \dots, x^p , onde a ordem com que estes elementos são apresentados é relevante. A **Figura 13** a estrutura de uma RNN simples, onde este processa um elemento da sequência por vez, o neurónio recursivo não leva em consideração apenas o presente elemento, mas também o estado (*output, cell state*) anterior deste neurónio como entrada. Assim,

a saída do RNN em um instante t é também uma função de todas as entradas anteriores. A parte do RNN que conserva algum estado interno ao longo dos passos temporais é chamado de **célula de memória** (Géron, 2019). A **Figura 13** ilustra a operação recursiva em uma RNN composto por um neurónio.

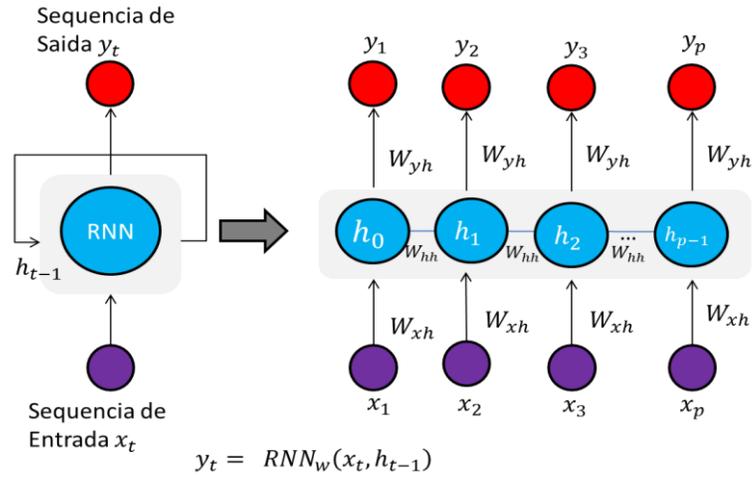


Figura 13: Neurónio Recursivo : um neurónio recursivo (esquerda) desenrolado através dos passos temporais (direita). **Fonte:** Adaptado de : (Brunton & Kutz, 2021)

Onde Y^i é usado para representar o *output*, e h^i para representar a memória do neurónio (também chamado de estado). Matematicamente, o procedimento para processar o i – ésimo elemento pode ser formulado como:

$$h^i = \alpha_h (W_{hh} * h^{i-1} + W_{hx} * X^{i-1} + b_h) \quad (2. 12)$$

$$y^i = \alpha_y (W_{hy} * h^i + b_y) \quad (2. 13)$$

Onde W_{hh} , W_{hx} e W_{hy} São as matrizes para realizar a transformação linear; b_y , b_h São os *bias*; e α_h e α_y São as funções de activação. Esta é estrutura RNN simples, para sequências muito longas, os RNNs simples tendem a ser instáveis, portanto aplicam-se estruturas mais sofisticadas como *Long Short-Term Memory* (LSTM) e sua variante: *Gated Recurrent Units* (Géron, 2019).

b. LSTM

Uma das primeiras e mais usadas soluções para conter o problema de desaparecimento e explosão de gradientes inerentes ao treinamento de redes neurais baseadas em unidades recorrentes simples vem na forma de **Long Short-Term Memory**, ou abreviadamente chamado de LSTM (Zhang et al., 2023).

Redes neurais LSTMs assemelham-se bastante as redes neurais recorrentes convencionais, mas aqui cada neurónio tradicional é substituído por células de memória com estado curto da célula (ou estado escondido (*hidden state*)) $h^{(t)}$ e estado longo da célula (ou estado da célula (*cell state*)) $C^{(t)}$ (Zhang et al., 2023). Cada célula de memória é equipada com um estado interno (*internal state*) e um número de portas que determina se: (i) o *input* deve alterar o estado interno (porta e entrada, *input gate*), (ii) o estado interno deve ser reduzido a zero (porta de esquecimento, *forget gate*), e o estado interno de um dado neurónio deve impactar o *output* da célula (porta de saída, *output gate*) (Zhang et al., 2023).

Este conjunto de portas (*gates*) que é aplicado para controlar o fluxo de informação dentro da célula LSTM representa a real diferença entre LSTMs e redes neurais recorrentes simples, ou seja, este tem um mecanismo dedicado para quando o estado da célula deve-se actualizar e também repor (*reset*). O funcionamento destas portas ou mecanismo é aprendido durante o treinamento e eles respondem as actividades destacada acima: armazenar, eliminar e ler informações do estado da célula $C^{(t)}$ (Zhang et al., 2023).

A informação fluindo entre passos temporais consecutivos de uma sequência dentro de LSTMs inclui o **estado de longo-termo da célula** (estado da célula, *long-term state* ou *cell state*) $C^{(t)}$ e o **estado curto da célula** (*short-term state*) $h^{(t)}$. O *cell state* $C^{(t-1)}$ funciona como a informação do instante anterior que é propagada para o próximo passo, e o *hidden* $h^{(t-1)}$ *state* auxilia na decisão de como a informação deve ser propagada e também serve de *output* para o instante t se necessário (Zhang et al., 2023). Conforme ilustrado na **Figura 14**.

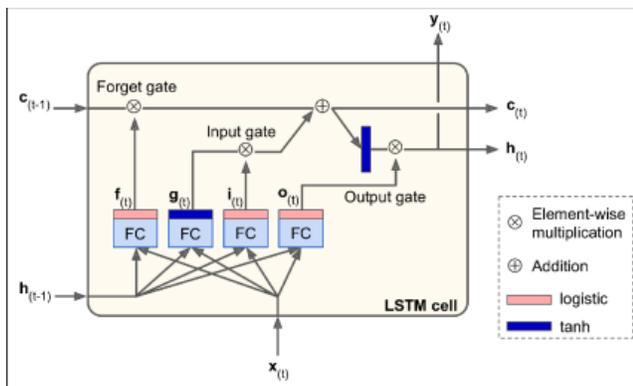


Figura 14: Unidade LSTM. **Fonte:** (Géron, 2019)

Conforme descrito, a primeira operação do LSTM é decidir qual informação do estado da célula anterior $C^{(t-1)}$ Será descartada. Esta decisão é feita pela porta de esquecimento (f_t), que é composta por uma função de activação sigmóide (*logistic*, σ) que retorna valores entre 0 e 1. Onde valor próximo (ou igual) de 0 significa que pouca informação do estado anterior será passada adiante, e valor próximo (ou igual) de 1 significa que muita informação do estado anterior será passada adiante. Analogamente funcionam a outras portas o_t e I_t .o *output* desta porta pode ser resumida como um vector f_t , Que tem a mesma dimensão que $C^{(t-1)}$:

$$f_t = \sigma (W_f * x^t + U_f * h^{(t-1)} + b_i) \quad (2.14)$$

A próxima operação é determinar a quantidade de informação do novo elemento x^t Será armazenada no novo estado da célula. Similarmente a *forget gate*, porta de entrada (I_t) é adicionada para realizar essa decisão :

$$I_t = \sigma (W_i * x^t + U_i * h^{(t-1)} + b_i) \quad (2.15)$$

A informação x^t é processada por uma rede neural para gerar um novo estado da célula candidato \check{c}_t , que é usada para actualizar o estado de célula. Ela poder resumida como:

$$\check{c}_t = \tanh(W_c * x^t + U_c * h^{(t-1)} + b_i) \quad (2.16)$$

A partir do estado candidato, gera-se o novo estado $C^{(t)}$ Combinando o estado anterior $C^{(t-1)}$ e o estado candidato \check{c}_t Como:

$$c^t = f_t \odot C^{(t-1)} + I_t \odot \check{c}_t \quad (2.17)$$

Por último, produz-se o *hidden state* $h^{(t)}$ que pode fluir para o próximo passo $t + 1$ ao mesmo tempo, serve de previsão Y'_t para o passo t Se necessário. O novo *hidden state* é baseado na actualização do estado da célula c^t pela porta de saída que determina que parte do estado da célula deve ser preservado. A porta de saída é formulada como:

$$o_t = \sigma(W_o * x^t + U_o * h^{(t-1)} + b_i) \quad (2.18)$$

E o estado interno como:

$$h^t = o_t \odot \tanh(c^t) \quad (2.19)$$

Desta forma, arquitetura geral de uma célula LSTM permite que a célula aprenda eliminar (f_t), Armazenar i_t e ler (o_t) Informações de sua longa memória c^t (Zhang et al., 2023).

A operação LSTM pode ser resumido como: $Y'_t, c^t, h^t = LSTM(x^{t-1}, c^{t-1}, h^{t-1})$ (2. 20)

Ambos Gated Recurrent Unit (GRU) e LSTM são variantes de neurónios neurais recursivos, exceto que unidades LSTM têm uma porta de entrada, saída, e uma porta de esquecimento, mas GRU tem uma porta de actualização e uma porta de reposição, o que lhes concede robustez no processamento de longas seqüências (Zhang et al., 2023).

2.3.4.4. Conexões Residuais e Blocos Residuais

Conexões residuais ou conexões de salto (*Residual connection* ou *Residual Blocks*) são ramificações no caminho computacional onde o *input* de cada uma das camadas $f[*]$ é adicionada de volta ao *output* da camada. **Figura 15** mostra esquematicamente um exemplo de conexão de salto.

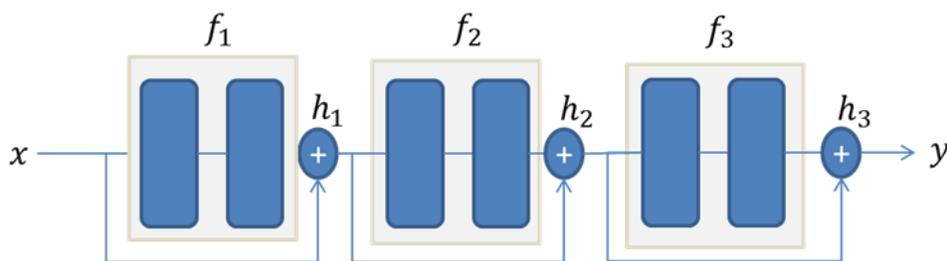


Figura 15: Conexões e Blocos Residuais. **Fonte:** Adaptado de (Zhang et al., 2023)

Matematicamente, estas operações são descritas como:

$$\mathbf{h}_1 = \mathbf{x} + \mathbf{f}_1[\mathbf{x}, \boldsymbol{\phi}_1] \quad (2. 21)$$

$$\mathbf{h}_2 = \mathbf{h}_1 + \mathbf{f}_2[\mathbf{h}_1, \boldsymbol{\phi}_2] \quad (2. 22)$$

$$\mathbf{h}_3 = \mathbf{h}_2 + \mathbf{f}_3[\mathbf{h}_2, \boldsymbol{\phi}_3] \quad (2. 23)$$

$$\mathbf{y} = \mathbf{h}_3 \quad (2. 24)$$

Onde o primeiro termo do lado direito das equações é a conexão de salto. Cada uma das f_k Funções uma mudança aditiva da corrente representação. Cada combinação aditiva da entrada e da saída é conhecido como bloco residual ou camada residual (Zhang et al., 2023).

2.3.5. Previsão de Séries Temporais com Métodos Supervisionados de Aprendizado de Máquinas

Sobe o ponto de vista matemático, uma serie temporal é uma serie de pontos de dados indexados (ou listados ou representados graficamente) em ordem temporal obtidas sequencialmente e igualmente espaçadas no tempo. Portanto, é uma sequência de dados em tempo discreto. A previsão de séries temporais é uso de um modelo para prever valores futuros da sequência com base nos valores observados anteriormente. Abordagens tradicionais de previsão de séries temporais incluem média móvel (*moving average*), suavização exponencial (*exponential smoothing*), etc. No entanto, métodos modernos incluem algoritmos supervisionados de ML, como redes neurais profundas de DL, especialmente RNNs (Brownlee, 2019).

Para aplicação de métodos de supervisionados de ML, como DL, as séries temporais precisam ser reestruturadas de tal modo que intervalos anteriores da serie representem o *input* (x) e o intervalos posteriores representem o *output* (y). O uso de intervalos de tempo anteriores para prever o valor do próximo passo temporal da sequência é chamado de método de janela deslizante (*sliding window*), mas também é chamado de método da janela ou método do atraso (*lag método*). O número de instantes anteriores dentro de uma janela determina a largura da janela (*window size*) ou tamanho do atraso (*lag size*) (Brownlee, 2019). Este procedimento é ilustrado na **Figura 16**.

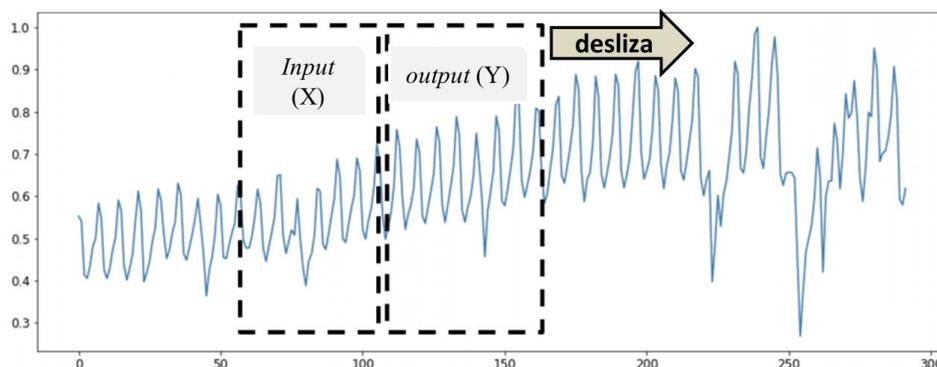


Figura 16: Método de deslizamento de janela usado para gerar os dados de X e Y de series temporais para o treinamento de modelos de Machine Learning.

De modo geral, o método de janela é base para transformação de um conjunto de dados de series temporais, como registos de trajetórias, em um problema supervisionado de ML, quer seja regressão ou classificação (Brownlee, Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs, and LSTMs in Python, 2019).

2.3.6. Modelos de Redes Neurais para previsão de trajetória de CTs

As redes neurais artificiais modelam as relações lineares e não-lineares entre as variáveis dependentes e independentes através dos dados observados do problema em causa. O modelo final é representado pela arquitectura e tipologia das redes definidas pelo usuário, e pelos coeficientes destas redes que são ‘aprendidos’ automaticamente através do treinamento repetitivo e interativo do modelo. Abaixo, são descritos algumas arquiteturas de rede propostas para este estudo:

2.3.6.1. Modelo Stacked-LSTM

O modelo Stacked-LSTM consiste de empilhamento de 3 camadas de unidades LSTM com 128, 64, e 32, respectivamente alocadas uma por cima da outra, resultando em uma rede recorrente profunda na direção do tempo e do output. Nesta composição, todos neurónios de uma camada a cada instante t , dependem da saída do instante anterior $t-1$ da mesma camada e do output da camada anterior no instante t . A estrutura esquemática do Stacked-LSTM é apresentada na Figura 17.

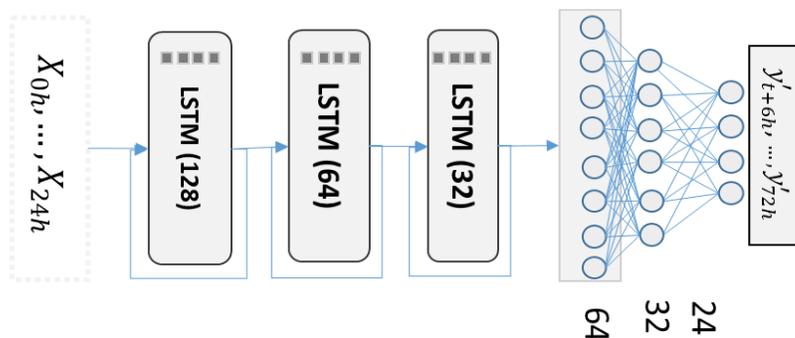


FIGURA 17: ARQUITETURA DA REDE STACKED-LSTM

As camadas de LSTMs lêem um passo temporal da sequência por vez construindo uma representação interna contextual extraída da sequência que é enviada para as 3 camadas de neurónios simples com 64, 32 e 24 neurónios, respectivamente onde são feitas as previsões de y_t dentro do horizonte de previsão.

2.3.6.2. Modelo BN-LSTM

De modo similar ao Stacked-LSTM, o BN-LSTM é uma rede neural recorrente profunda feita de unidades LSTM. No entanto, o BN-LSTM é mais profundo com alguns componentes especiais, nomeadamente: camada de *Batch Normalization*² e camada de *Dropout*³. A adição destas camadas permite a construção e treinamento de redes mais profundas evitando o *overfitting*, e explosão ou desaparecimento de gradientes (*gradient vanishing ou explosion*) durante o treinamento da rede.

Figura 18 ilustra a estrutura esquemática da rede/modelo BN-LSTM construído através do empilhamento de camadas de unidades LSTMs combinadas a camadas *Batch Normalization* e *Dropout*.

² *Batch Normalization* é uma técnica introduzida por (Ioffe & Szegedy, 2015) para reduzir o perigo do desaparecimento e explosão de gradientes no treinamento de redes neurais profundas

³ *Dropout* é uma das técnicas de regularização mais popular introduzida por (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012), e posteriormente detalhada por (Srivastava, 2014) para reduzir a complexidade e evitar o fenómeno de *overfitting* em redes neurais profundas

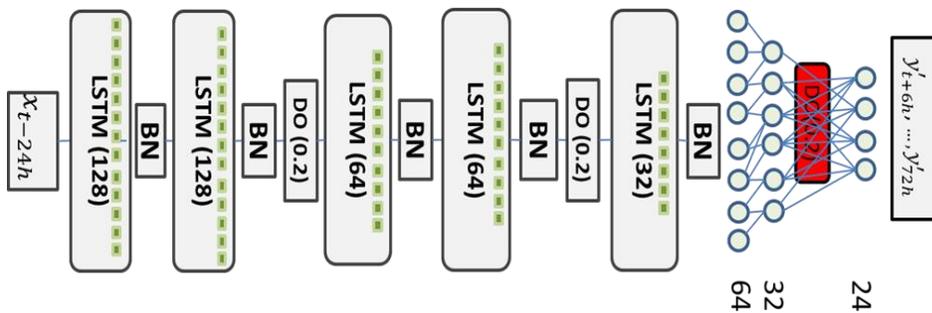


FIGURA 18: ESTRUTURA DO MODELO BN-LSTM

O modelo LSTM é composto por 5 camadas de dezenas de neurónios LSTM intercaladas por camadas de *Batch Normalization*: 128, 128, 64, 64, e 32 LSTMs, respectivamente. A transição de 128 a 64, e de 64 a 32 LSTMs é intercalada por uma camada de *Dropout* com rácio 20% de *dropout*. Finalmente, os *outputs* do bloco de LSTMs são enviadas para uma camada densa de neurónios que produzem as finais previsões de δlat e δlon para as próximas 72 horas.

2.3.6.3. Modelo Skip-LSTM

Similarmente a outros modelos, o modelo SKIP-LSTM é uma rede profunda de neurónios recorrentes LSTM. No entanto, diferentemente de outras arquiteturas, o SKIP-LSTM introduz apresenta Conexões residuais.

Este tipo de arquitetura permite que redes mais profundas sejam treinadas de maneira estável. No entanto, as conexões de salto causam um aumento exponencial da magnitude das ativações. Portanto, para resolver tal efeito adicionam-se camadas de *Batch Normalization* nos blocos residuais. A **Figura 19** ilustra a estrutura esquemática (simplificada) do modelo *SKIP-LSTM* desenvolvido para elaboração deste trabalho.

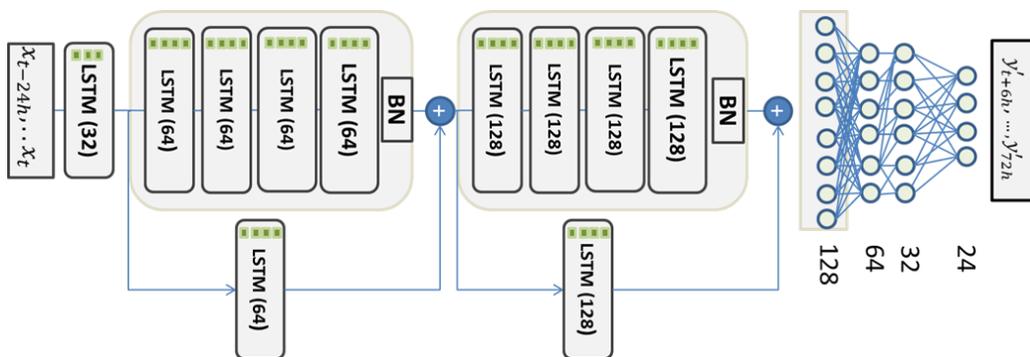


FIGURA 19: ESTRUTURA ESQUEMÁTICA DO MODELO SKIP-LSTM

O modelo *SKIP-LSTM* compreende dois blocos residuais compostos por uma camada de *Batch Normalization* e 4 camadas de neurónios LSTMs com 64 e 128 neurónios por camada, respectivamente.

Para além destas camadas, a rede neural tem uma camada de 32 neurónios LSTMs logo a seguir da camada de entrada. Por outro lado, as conexões de salto dos dois blocos residuais têm uma camada de 64 e 128 neurónios, respectivamente. Para que as previsões finais sejam feitas, o output do último bloco residual, ou seja, o contexto é enviado para 4 camadas densas de 128, 64, 32 e 24 neurónios simples, respectivamente, onde a última cada (24) é que produz as reais previsões.

2.3.6.4. Modelo CONV-GRU

O modelo Conv-GRU, similarmente a outros modelos, é uma rede profunda de neurónios neurais artificiais. A arquitetura da rede utiliza CNNs com Bloco Residuais, camadas de *Batch Normalization* e *Dropout*, activação Leaky-relu e Bidirecional Gated Recurrent Units (uma variação de neurónios GRU), mecanismo de atenção⁴ e rede de neurónios simples.

A **Figura 20** ilustra a estrutura esquemática dos componentes e da rede Conv-GRU. Para a implementação das operações feitas pelas Redes Neurais Convulsionais em Bloco Residuais introduzem-se módulos. Módulos são usados para facilitar implementação de redes complexas. Neste caso, o Resblock é um conjunto de operações convulsionais com conexões de salto feitos nos dados.

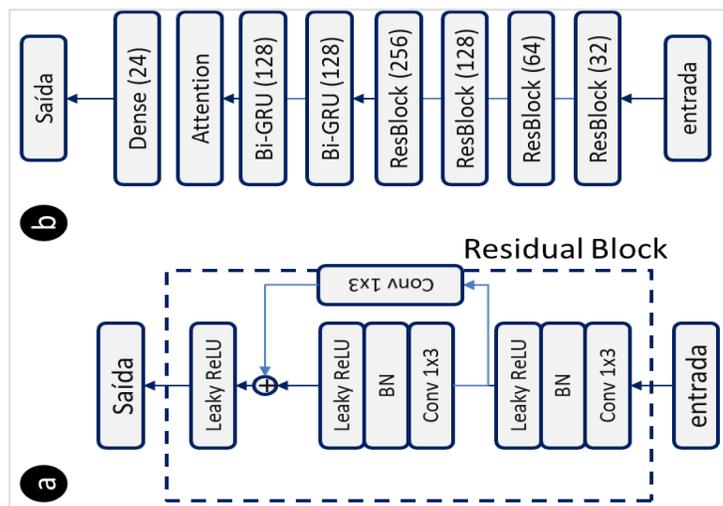


FIGURA 20: ESTRUTURA ESQUEMÁTICA DO MODELO CONV-GRU: A - BLOCO RESIDUAL (RESBLOCK) E B - ESTRUTURA COMPLETA DO MODELO

A rede final é composta por 5 blocos residuais de 32, 64, 128 e 256 neurónios 1D-conv, respectivamente; os padrões extraídos pelos blocos residuais são enviados para duas camadas consecutivas de Bidirecional

⁴ É uma técnica usada para permitir que as redes possam determinar parte da sequência de *input* é relevante para realizar previsões da sequência de *y* no instante *t* (Bahdanau, Cho, & Bengio, 2014)

Gated Recurrent Units (Bi-GRU) de 128 neurónios GRU ambas. Esta por sua vez é acompanhada por uma camada de atenção e uma camada densa de 24 neurónios responsável pelas previsões.

Os hiperparâmetros dos modelos são resumidos na **Tabela 6** onde a função de custo (*loss function*), tamanho do lote (*batch size*), taxa de aprendizagem (*learning rate*), e o método de treinamento (o otimizador).

2.3.6.5. Modelo ENCODER-DECODER

A rede *Encoder-Decoder*, similarmente aos outros modelos, é uma rede neural recorrente profunda. No entanto, ela compreende 2 módulos: um codificador (*encoder*) e um decodificado (*decoder*) acompanhado de um mecanismo de atenção.

Os módulos codificador e decodificador são essencialmente compostos de (múltiplas) camadas de (múltiplos) neurónios recorrentes. No caso do modelo criado neste trabalho, o codificador e decodificador são compostos por unidades LSTMs, onde os dados de toda sequencia da trajetória das anteriores 24 do ciclone são comprimidas em um estado interno (contexto) que é enviado para o decodificador que produz todas as previsões de ($\delta lat, \delta lon$) para todos os passos temporais dentro do horizonte de previsão (6-72h). A Figura 21 é uma representação simplificada da estrutura esquemática do modelo Encoder-Decoder.

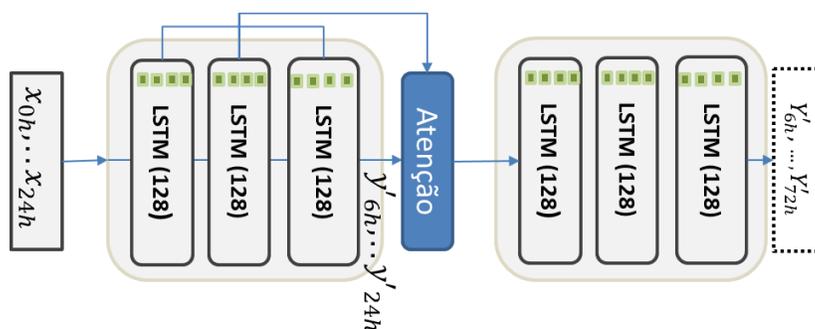


FIGURA 21: ESTRUTURA ESQUEMÁTICA DO MODELO ENCODER-DECODER

A arquitetura Encoder-Decoder é composta por 3 camadas de redes neurais de 128 neurónios LSTM por cada módulo (tanto para o codificador como o decodificador). O *decoder* não só recebe o estado interno final do codificador, este também recebe todas as saídas do codificador e, através do mecanismo de atenção Bahdanau (Bahdanau et al., 2014) determina qual parte desta informação é relevante para fazer a previsão de y no instante t sem que o encoder precise comprimir o sinal de entrada num estado final fixo (Zhang, Lipton, Li, & Smola, 2023)

TABELA 6: HIPERPARÂMETROS DE CONFIGURAÇÃO DOS MODELOS

MODELO	HIPERPARÂMETROS	VALORES
Stacked-LSTM	Método de treinamento (otimizador)	Adam
	Número de camadas LSTM	3
	Número de neurónios LSTM p/ camada	128, 128, 64
	Número de camadas Densas (camada de <i>output</i>)	3 (32,16, 24)
	<i>Learning rate</i>	5×10^{-3}
	<i>Learning rate schedule</i>	<i>ExponentialDecay</i>
	<i>Batch size</i>	32
	<i>Loss function</i>	Erro Médio Quadrado (<i>Mean squared error (MSE)</i>)
BN-LSTM	Método de treinamento	Adam
	Número de camadas LSTM	4
	Número de neurónios LSTM p/ camada	64, 128, 64, 32
	Número de camadas Densas (camada de <i>output</i>)	3 (64, 32, 24)
	Camadas Especiais	<i>Batch Normalization, dropout</i>
	<i>Learning rate</i>	5×10^{-3})
	<i>Learning rate schedule</i>	<i>ExponentialDecay</i>
	<i>Batch size</i>	32
	<i>Loss function</i>	Erro Médio Quadrado (<i>Mean squared error (MSE)</i>)
Skip-LSTM	Método de treinamento	Adam
	Número de camadas LSTM	9
	Número de neurónios LSTM p/ camada	1x 32, 4 x 64, 4 x 128

	Número de camadas Densas (camada de <i>output</i>)	
	Camadas Especiais	<i>Batch Normalization, Skip connections</i>
	<i>Learning rate</i>	<i>Default</i>
	<i>Learning rate schedule</i>	<i>None</i>
	<i>Batch size</i>	32
	<i>Loss function</i>	<i>MSE</i>
Conv-GRU	Método de treinamento	Adam
	Tipos de neurónios	GRU, CONV
	Número de camadas ResBlock	4
	Número de neurónios ResBlock camada	32, 64, 128, 256
	Número de camadas e neurónios GRU	2 x 128
	Número de camadas Densas (camada de <i>output</i>)	1 (24)
	Camadas Especiais	<i>Batch Normalization, Skip connections, attention</i>
	<i>Learning rate</i>	<i>Default</i>
	<i>Learning rate schedule</i>	<i>None</i>
	<i>Batch size</i>	32
	<i>Loss function</i>	<i>MSE</i>
ENCODER-DECODER	Método de treinamento	Adam
	Tipos de neurónios	LSTM
	Número de camadas e no Encoder	3 x 128
	Número de camadas e neurónios no Decoder	3 x 128
	Número de camadas Densas (camada de <i>output</i>)	1 x 1

Camadas Especiais	<i>Batch Normalization e attention</i>
<i>Learning rate</i>	<i>Default</i>
<i>Learning rate schedule</i>	None
<i>Batch size</i>	32
<i>Loss function</i>	<i>MSE</i>

2.4. Estudos Relacionados

Vários estudos foram feitos em algumas bacias com actividade ciclónica ao redor do mundo demonstrando o uso de DL na para previsão de trajetória de TCs. Um exemplo disto é o estudo feito por (Alemany et al., 2019) usando redes neurais recorrentes (RNNs) para prever Trajetória do TCS baseada em dados históricos de trajetórias de CTs. Por outro lado, Redes Neurais Convolucionais (CNNs) têm sido aplicadas para processar imagens de mapas reanálise e dados de trajetória para previsão de trajetória como por exemplo o estudo feito por (Giffard-Roisin, et al., 2020) no oceano Pacífico e Atlântico. No entanto, estudos recentes têm explorado a combinação de diferentes arquiteturas de redes (CNNs e RNNs) e fontes de dados, resultando em abordagens multimodais, para prever trajetórias de CTs, como o estudo feito por (Wang et al., 2023) no noroeste do oceano pacífico

Embora estes estudos, que se viram de inspiração para este principalmente (Wang et al., 2023), mostrem que a aplicação de DL para modelar CTs em diversas bacias seja promissória, poucos estudos foram feitos na bacia do SOI. Portanto, o presente estudo visa explorar o uso de redes neurais para prever trajetória de CTs sobre SOI, visto que esta é uma região vulnerável a sistemas de tempestade tropical como CT.

Capítulo 3 : Matérias e Métodos

Do ponto de vista metodológico, este trabalho pode ser dividido em duas fases principais, nomeadamente:

- ❖ **Fase de Pré-processamento:** esta é fase na qual algumas variáveis são criadas a partir dos dados fornecidos pelo *Best Track Archive for Climate Stewardship* (IBTrACKS), e de seguida, os dados são escalados, e por fim algumas variáveis do conjunto são seleccionadas para a segunda fase de acordo com o seu grau de importância para o problema;
- ❖ **Fase de Treinamento:** nesta fase, diferentes redes neurais são criadas e treinadas, e por fim comparadas para determinar o (s) modelo (s) com as melhores previsões, em média, dentre as redes propostas. De modo geral, o trabalho apresenta a previsão de séries temporais (posições de TCs) baseado em DL, especialmente redes neurais de neurónios recursivos.

3.1. Materiais

3.1.1. Coleção de Dados

Os dados de trajectória usados neste trabalho foram retirados do IBTrACKS. Os registos feitos pelo IBTrACKS abrangem todos os Ciclones Tropicais (ou tempestades) a nível global. Para cada CT, são dados a Latitude, Longitude, Pressão do Centro, Velocidade máxima do vento, Direcção, Velocidade de Deslocamento entre outros parâmetros em intervalos de 3 horas de tempo.

Neste trabalho foram seleccionados, do IBTrACKS, CTs que ocorreram sobre a Bacia do SOI no período 1993 a 2023⁵. Com base nestes dados foram retirados 6 variáveis (por cada CT), nomeadamente a Latitude, Longitude, Pressão do centro, Velocidade máxima do vento, Direcção e Velocidade de deslocamento, com base nas quais foram derivadas 19 variáveis (de trajectória e meteorológicas), mostradas na Tabela 7.

TABELA 7: LISTA DE VARIÁVEIS DE ESTUDO

NOME COMPLETO (DESCRIÇÃO)	ABREVIACÃO
Ângulo formado entre os momentos passados e o momento actuais	ANG
Longitude dos momentos passados	LON

⁵ No processo de limpeza, foram eliminados sistemas que não atingiram o estágio de Tempestade Tropical segundo a escala usada pela Météo-France **Tabela 3** durante parte de sua vida, assim também como foram eliminados sistemas com faltas e erros de registo (inconsistência temporal).

Diferença de direcção entre os momentos passados e os momentos actuais	DIR_DIF
Distância meridional entre os momentos passados e o momento actuais	MER_DIS
Diferença de longitude entre os momentos passados e o momento actual	LON_DIF
Latitude dos momentos passados	LAT
Diferença de velocidade de deslocamento entre os momentos passados e o momento actual	MOV_SP
Parâmetros de Coriolis dos momentos passados	COR
Distância zonal entre os momentos passados e o momento actual	ZON_DIS
Pressão atmosférica central dos momentos passados	PRE
Velocidade máxima do vento dos momentos passados	WIND_SP
Diferença de latitude entre os momentos passados e o momento actual	LAT_DIF
Distância de deslocamento entre os momentos passados e o momento actual	DIS
Velocidade de deslocamento zonal do momento passado	ZON_SP
Velocidade de deslocamento meridional do momento passado	MER_SP
Direcção de deslocamento dos momentos passados	MOV_DIR
Diferença de pressão atmosférica entre os momentos passados e o momento actual	PRE_DIF
Diferença de velocidade máxima do vento entre os momentos passados e o momento actual	WIND_SP_DIF
Velocidade de deslocamento dos momentos passados	MOV_SP

3.2. Métodos

3.2.1. Pré-Processamento de Variáveis

O primeiro procedimento de pré-processamento aplicado aos dados antes prosseguir com o treinamento dos modelos foi dimensionamento de variáveis. Dentre os procedimentos de dimensionamento disponíveis, no presente aplicou-se a padronização⁶ dos valores das variáveis.

O segundo pré-processamento foi a selecção das variáveis mais descritivas (que melhor explicam a variância de *output*) do problema. Para que tal fosse alcançado, foi usado o modelo *Random Forest Regressor*⁷ (Regressor de Floresta Aleatória) sobre os dados. Classificando as variáveis de acordo com

⁶ Este método implica que valores padronizados tenham média igual a zero e variância unitária (Géron, 2019).

⁷ *Random Forest regressor* é um conjunto (*Ensemble*) de árvores de decisão aplicados para criar modelos regressão (Géron, 2019).

o seu grau de importância, o *Random Forest Regressor* (RFR) seleciona a melhor combinação de variáveis para o problema e, portanto, permite a seleção dos recursos (variáveis) dos dados mantendo apenas a melhor combinação variáveis de acordo com o impacto que cada variável tem sobre precisão das previsões, e de seguida, estes dados foram usados para treinar as redes neurais artificiais descritas na seção Modelos de Redes Neurais para previsão de trajetória de CTs

3.2.2. Implementação de Redes Neurais Artificiais

Neste estudo, o autor propôs várias arquiteturas de redes, com cada uma das redes capturando diferentes aspectos espaço-temporais dos dados de acordo com a estrutura. Tais redes foram propostas com intuito de explorar cada vez mais a eficácia de redes neurais artificiais para modelar a trajetória de CTs indo de redes mais simples a redes mais complexas. Estas redes são essencialmente baseadas em neurónios recursivos, especialmente LSTM e GRU. Conforme descrito na seção [Modelos de Redes Neurais para previsão de trajetória de CTs], para o experimento, cada modelo de redes neurais foi treinado com um máximo de 200 épocas (*epochs*) com *early-stopping* configurado para automaticamente encerrar o treinamento se o error medido nos dados de validação não reduzir após 10 épocas de treinamento seguidas e, salvar a última melhor configuração dos parâmetros (*weights*) da rede sendo treinada.

Devido a limitações computacionais, não foram feitos vários testes com várias variações de hiperparâmetros (*hyperparameters*) destas redes, por exemplo, diferentes números de camadas ou neurónios por camada destes modelos. O treinamento de cada modelo levou em média 45 minutos.

❖ Medidas de Performance

O error médio absoluto (*mean absolute error*, MAE), a raiz do erro médio quadrado (*root mean squared error*, RMSE), e Error da distância (em quilómetros), ED, foram escolhidas como as medidas para comparar a performance dos modelos na Previsão das trajetórias. As expressões matemáticas correspondentes são apresentadas nos Anexos.

3.2.3. Reconstrução de Trajectórias

Os modelos não foram treinados para prever diretamente a posição do centro CT a cada instante, ao invés disso, estima-se o deslocamento (δ) da latitude (e longitude) em relação ao presente momento e o instante anterior para cada instante no futuro. Portanto, para construir as posições, adiciona-se a latitude (e longitude) do presente momento ao deslocamento ($\delta_{lat}, \delta_{lon}$) do instante $t + 1$ previsto pelo modelo.

Todos os Modelos e operações, desde o Pré-Processamentos à reconstrução de trajetórias aqui descritas, foram implementadas através da linguagem de programação Python (Python Software Foundation, 2019).

3.2.4. Comparação de trajetórias e Contrução do Ensemble

Para demonstrar a aplicação prática dos modelos produzidos neste estudo, realizou-se uma comparação entre a performance destes modelos e o modelo actualmente usa pelo Meteo-France na Lá-Reunion. Para tal, foram selcionados alguns ciclones para verificar quão próximo as trajetórias de cada um dos modelos pode estar das reais observações.

Visto que cada um dos modelos apresenta propriedades diferentes, assim como diferentes habilidades de previsão, os modelos seleccionados foram combinados através do método de aprendizam designado por *Ensemble Learning*⁸. O *Ensemble Method* usado para este fim designa-se por *Stacked Generalization Ensemble*⁹ que consiste no treinamento de um modelo de aprendizagem para ‘aprender’ a combinar previsões feitas por diferentes modelos habilidosos em uma única previsão a cada instante de execução (Brownlee, 2019). Para construção deste modelo de combinação geralmente usam-se modelos lineares para garantir a simplicidade, por essa razão usou-se o modelo *stochastic gradient descent regressor*, que é um modelo linear baseado na optimização através do algoritmo GD descrito na secção [2.3.2].

⁸ Em machine Learning, um grupo de predictores (modelos) é chamado *Ensemble*, assim, está técnica de previsão através da combinação de modelos é chamado de *Ensemble Learning* e, o algoritmo de *Ensemble Learning* é chamado de *Ensemble Method*. Esta técnica de produção de modelos de ML é usado para reduzir a variância das previsões de Modelos sigulares (Géron, 2019).

⁹ Estes modelos possuem dois níveis de processamento, nomeadamente Nível 0 – os dados do nível 0 são os *inputs* dos dados de treinamento (validação ou teste) e os modelos de nível 0 fazem previsões deste dados e; Nivel 1- os dados do nível 1 tomam os outputs dos modelos de nível 0 como *inputs* e o único modelo de nível 1 (aprende) faz previsões deste dados (Brownlee, 2019)

Capítulo 4 : Resultados e Discussão

4.1. Seleção de Variáveis

A **Figura 22** (a) mostra as 19 variáveis ordenadas pelo seu índice de importância (quanto maior o índice, maior é a importância da variável) estimados por meio do RFR. As pontuações OOB¹⁰ de diferentes dimensões de variáveis de entrada são calculadas e mostrados na **Figura 22** (b) com as variáveis ordenadas por importâncias. Neste caso, as primeiras onze (11¹¹) variáveis mostram-se mais dominantes para prever a mudança (deslocamento) de latitude e longitude dentro das próximas 72 horas, com as características como a longitude e ângulo das horas anteriores sendo mais significativas. O gráfico também sugere que a adição de mais variáveis além das primeiras onze não altera muito a qualidade do modelo, o que indica que manter apenas 11 variáveis é suficiente para produzir um modelo de qualidade.

- **Criação das sequências de *input e output***

O objetivo deste trabalho é prever o movimento de CTs para as próximas 6-72 horas usando os dados da trajetória das anteriores 24 horas como *input* das redes neurais artificiais. Para tal, são explorados movimento de CTs na bacia do SOI de 1993 a 2023 e considerando as mudanças de latitude e longitudes ($\delta_{latitude}$, $\delta_{longitude}$) nas próximas 6-72 horas como variáveis quantitativa de previsão, com o centro dos CTs no presente momento sendo o ponto de referência. Neste estudo foram considerados 326 tempestades que geraram 12941 amostras através da aplicação do método de janela *sliding window* sobre a serie temporal das posições dos CTs. Estas amostras são divididas s em 3 grupos percentuais: 90 % - dados para treinamento (dos quais, 10 % - validação) e 10 % para teste. As informações detalhadas dos dados do estudo são resumidas pela **Tabela 8**.

¹⁰ Já que amostragem com reposição (*Bagging*) é usado para gerar as árvores de decisão algumas amostras não ‘vistas’ (*out-of-bag samples*) pela RFR durante o treinamento das árvores. Através destas amostras, pode ser avaliada a precisão do RF, obtendo-se as pontuações OOB (Géron, 2019)

¹¹ ANG, LON, DIR_DIF, MER_DIS, LON_DIF, LAT, MOV_SP, COR, ZON_DIS, PRE, WIND_SP

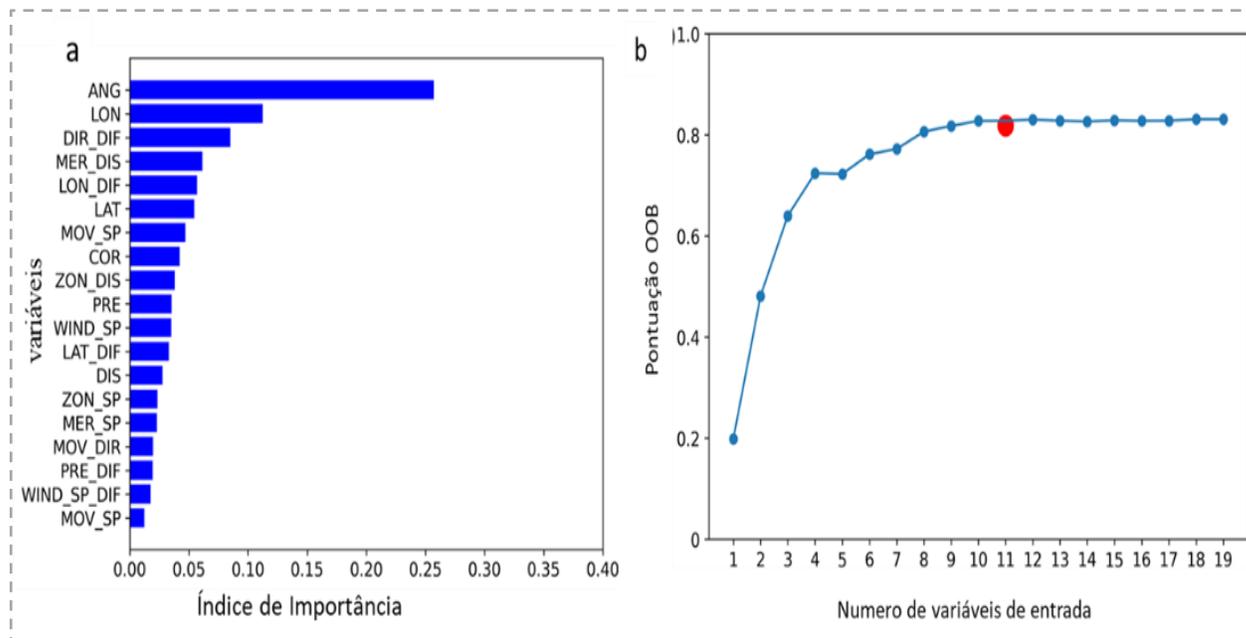


FIGURA 22: (A) O ÍNDICE DE IMPORTÂNCIA DAS VARIÁVEIS E (B) A PONTUAÇÃO OOB DE DIFERENTES COMBINAÇÕES DE VARIÁVEIS.

TABELA 8: DETALHES DOS DADOS EXPERIMENTAIS

PARÂMETROS	VALORES (DESCRIÇÃO)
Abrangência Temporal do Conjunto de Dados	De 1 de Janeiro de 1993 a 31 de Dezembro de 2023; 326 CTs
Tipo de Sistemas Tropicais	Todos sistemas que pelo menos atingiriam estágio de Tempestades Tropical
Dados de Treinamentos	10 481
Dados de Validação	1165
Dados de Teste	1295
Dimensão Temporal	3 Horas por cada registro
Dados Passados	24 Horas anteriores ao presente instante
Horizonte de Previsão	72 Horas após o presente instante

De seguida, amostras de dados de treinamento com oito passos temporais, geradas a partir do *sliding window* sobre cada CT, e 11 variáveis selecionadas através do método *Random Forest* de acordo com a sua importância [4.1.], foram usadas para treinar 5 arquiteturas de redes neurais recorrentes (modelos) [2.3.6] propostos para prever 12 passos da posição do CT em termos do deslocamento da latitude e longitude do CT (δlat , δlon) dentro de 6-72h. Para melhor avaliar e, conseqüentemente, selecionar o(s)

modelo(s) de destaque através de cada uma das medidas de performance [3.2.2], estes modelos foram avaliados nos dados de treinamento e de teste, conforme segue:

4.2. Análise dos treinamentos

O custo (*cost* ou *loss*) de treinamento e validação de todas arquiteturas de modelos propostos são apresentados nas **Figura 23** e **Figura 24**.

Estas as curvas informam a estabilidade, convergência e o potencial de generalização de cada modelo. Portanto, destes gráficos é visível que os modelos Stacked-LSTM (Figura 23a), Conv-GRU (Figura 24 c) e o ENCODER-DECODER (Figura 24 d) atingiram os menores valores do RMSE dentre os modelos. No entanto, o modelo Conv-GRU (Figura 24 c) e o ENCODER-DECODER (Figura 24 d) apresentam uma aprendizagem mais suave com a curvas de treinamento e validação extremamente próximas. O que sugere que estes foram bem treinados. Por outro lado, o modelo Stacked-LSTM apresenta curvas de treinamento e validação ligeiramente afastadas uma da outra, o que indica que o modelo Stacked-LSTM (23a) tenha sofrido de overfitting.

Para o caso dos modelos BN-LSTM (Figura 23 b) e SKIP-LSTM (Figura 24 e), estes atingiram valores RMSE de relativamente maiores. No entanto, destes dois modelos, o modelo BN-LSTM destaca-se por atingir valores RMSE relativamente baixos embora a curva da validação seja acidentada. Por outro lado, o modelo SKIP-LSTM (Figura 24 e) tem as piores curvas de aprendizagem com a curva de validação e treinamento extremamente afastadas. Esta configuração das curvas do modelo SKIP-LSTM tenha sofrido de underfitting.

De modo geral, estes gráficos informam que os modelos Stacked-LSTM, Conv-GRU e o ENCODER-DECODER superaram os modelos BN-LSTM e SKIP-LSTM durante o treinamento e, portanto, os modelos Stacked-LSTM, Conv-GRU e o ENCODER-DECODER são mais prováveis de ter uma melhor performance e generalização no teste.

Embora os gráficos da aprendizagem sejam informativos sobre a qualidade dos modelos, eles não são suficientes para tomar qualquer decisão sobre qual modelo escolher em detrimento do outro. Portanto é relevante reportar a qualidade dos modelos no conjunto de teste para garantir que se seleciona o modelo com a melhor performance e generalização sobre dados não antes vistos pelos modelos.

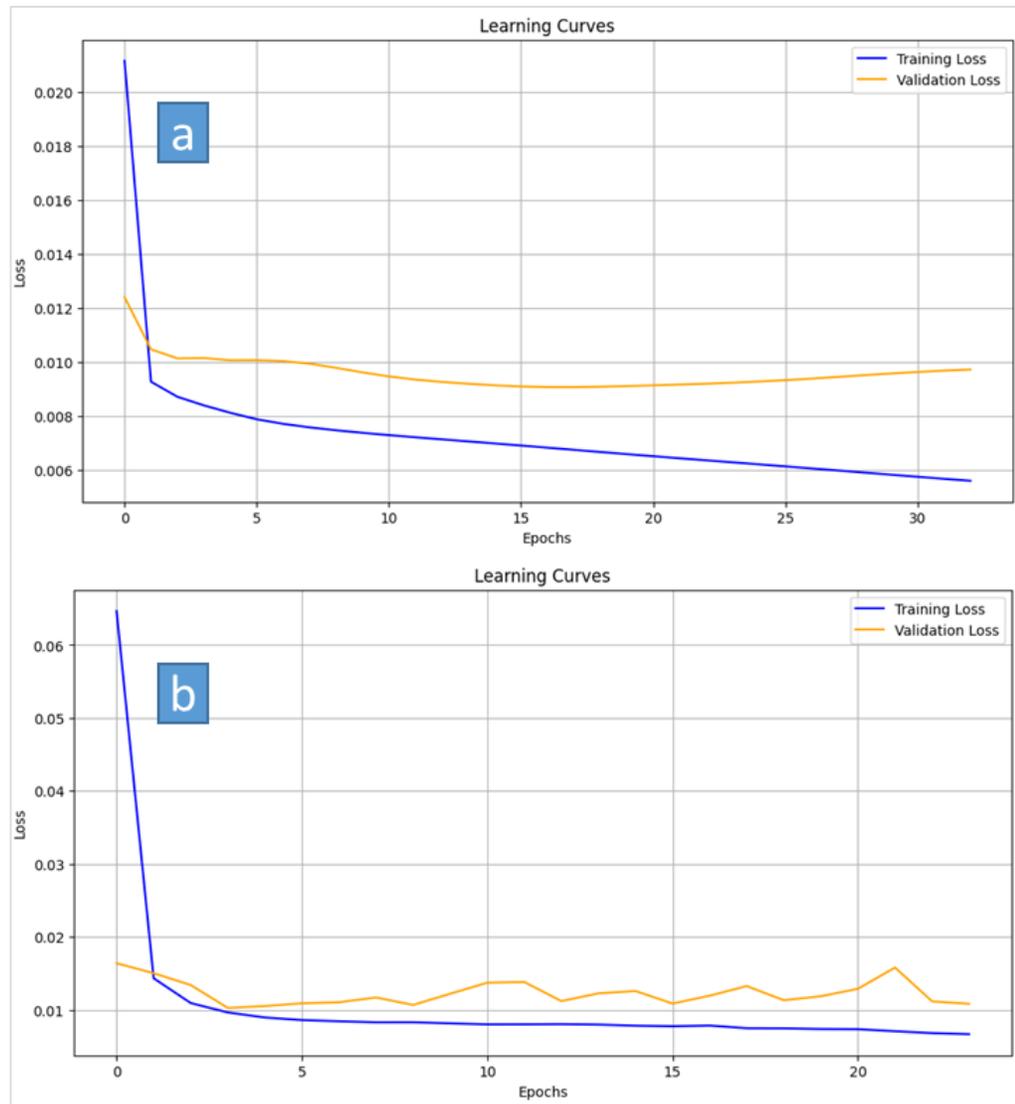


Figura 23: Curvas de Aprendizagem: a - Modelo Stacked-LSTM, b - BN-LSTM

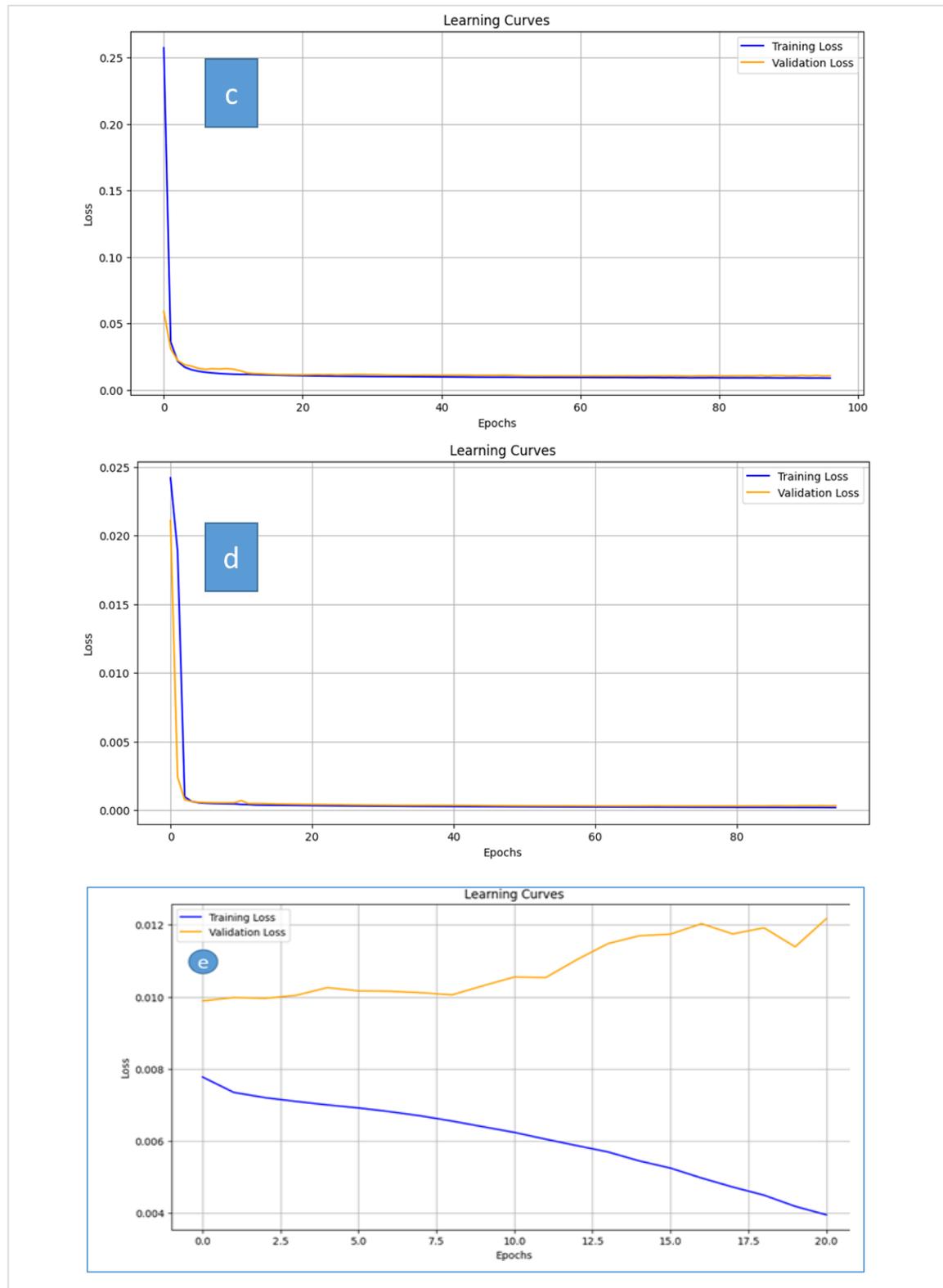


FIGURA 24: C - CONV-GRU, D - ENCODER - DECODER, E - SKIP-LSTM

4.3. Análise da Performance dos modelos nos dados de Teste

Tabela 9 e Tabela 10 mostram médias do erro dos modelos em termos do RMSE e MAE no conjunto de teste, respectivamente, e permitem a análise da qualidade dos modelo através dos valores obtido de *RMSE* e *MAE* entre os valores reais e estimados de $(\delta lat, \delta lon)$ pelos modelos. Onde os melhores modelos estão destacados a negrito.

TABELA 9: ERRO DE PREVISÃO DOS MODELOS EM TERMOS DE RMSE

$\delta(LATITUDE)$						$\delta(Longitude)$				
Tempo de Previsão						Tempo de Previsão				
	6h	12h	24h	48h	72h	6h	12h	24h	48h	72h
Stacked-LSTM	0.193	0.441	0.905	1.904	2.957	0.242	0.548	1.233	2.964	4.998
BN-LSTM	0.244	0.482	1.012	2.005	3.333	0.300	0.621	1.348	2.706	5.478
LSTM-SKIP	2.184	3.892	6.580	10.203	22.926	2.460	4.360	8.250	14.008	20.422
CONV-GRU	0.223	0.463	0.968	1.757	3.112	0.295	0.627	1.361	2.635	5.245
ENCODER-DECODER	0.137	0.349	0.856	1.716	3.659	0.155	0.440	1.158	2.447	5.193

TABELA 10: ERRO DE PREVISÃO DOS MODELOS EM TERMOS DE MAE EM GRAUS

$\delta(LATITUDE)$						$\delta(Longitude)$				
Tempo de Previsão						Tempo de Previsão				
Modelos	6h	12h	24h	48h	72h	6h	12h	24h	48h	72h
Stacked-LSTM	0.134	0.340	0.513	0.634	1.141	0.137	0.416	1.300	3.410	3.965
BN-LSTM	0.156	0.266	0.394	0.663	1.407	0.149	0.431	1.257	2.835	3.406
LSTM-SKIP	2.246	4.130	7.020	12.37	22.55	2.205	3.739	7.267	15.43	21.88
CONV-GRU	0.155	0.327	0.479	0.710	1.512	0.132	0.373	1.181	3.668	5.595

ENCODER- DECODER	0.106	0.272	0.658	1.574	2.902	0.114	0.324	0.876	2.252	3.864
---------------------	--------------	-------	-------	-------	-------	--------------	--------------	--------------	--------------	-------

Da **Tabela 9** verifica-se que o modelo ENCODER-DECODER supera todos os outros modelos na previsão de $(\delta lat, \delta lon)$ nos primeiros instantes do horizonte de previsão, tendo os menores valores de RMSE. Logo depois segue de Stacked-LSTM e Conv-GRU. Isto significa que o ENCODER-DECODER lidera a lista dos modelos de 6- 42h. No entanto, no intervalo posterior o modelo Stacked-LSTM apresenta melhoria significativa nos valores de RMSE.

Enquanto a **Tabela 9** permite facilmente discriminar o (s) modelo (s) através do número de vezes que se destacam que se destacam, a **Tabela 10** mostra um padrão de valores menores um pouco distribuído ao longo de cada modelo. Mas de modo geral, os resultados não se diferem muito do esperado das curvas de aprendizagem, e também do disposto na **Tabela 9**.

Os valores de RMSE e MAE obtidos pelos modelos sugerem que os modelos são mais robustos na previsão da latitude em relação a longitude, onde cada modelos apresenta valores menores de δ (Latitude) em cada das tabelas (RMSE e MAE) em relação aos valores de δ (Longitude).

Uma visão mais intuitiva da performance dos modelos é fornecida **Tabela 11**, ou seja, o erro da distancia média de cada modelo. Esta informação nos permite verificar, em quilômetros, o quão distante a posição prevista está da posição real do centro.

TABELA 11 COMPARAÇÃO DO ERRO MÉDIO DA DISTÂNCIA (EM KM) PRODUZIDOS POR CADA UM DOS MODELOS PROPOSTOS NO TRABALHO

MODELS	PREDICTION TIME							
	6h	12h	18h	24h	30h	36h	42h	48h
Stacked-LSTM	28.09	64.01	100.60	139.83	180.75	224.38	272.74	321.21
BN-LSTM	34.98	71.97	116.24	155.59	200.86	257.07	313.71	355.72
LSTM-SKIP	352.16	622.38	870.35	1110.57	1307.14	1583.32	1798.40	1955.18
CONV-GRU	34.10	71.11	111.24	153.18	197.14	242.12	288.29	335.78
ENCODER- DECODER	19.01	51.04	88.69	130.57	175.93	224.16	274.77	326.41

Modelos	Tempo de Previsão
---------	-------------------

Previsão de Trajectória de Ciclones Tropicais no Sudoeste do Oceano Índico Baseado em Modelos de Aprendizado Profundo

	42h	48h	54h	60h	66h	72h
Stacked-LSTM	272.74	321.21	363.82	419.65	468.56	468.56
BN-LSTM	313.71	355.72	414.02	465.01	524.40	587.74
LSTM-SKIP	1798.40	1955.18	2467.81	2249.89	2467.81	3167.09
CONV-GRU	288.29	335.78	384.78	435.95	488.98	543.93
ENCODER-DECODER	274.77	326.41	379.99	438.10	502.12	569.64

A **Tabela 11**, em conformidade com as tabelas anteriores, indica que o modelo ENCODER- DECODER supera todos os outros modelos, principalmente nas primeiras horas de previsão. No entanto o modelo Stacked-LSTM passa a ter a melhor performance a partir das 42 horas em diante, indicando que o modelo Stacked-LSTM é mais robusto na conversação, e propagação do contexto anterior da trajetória (*input*) ao longo do tempo pelo facto de ter o erro relativamente menor em todos os passos temporais. A **Figura 25** mostra que a relação entre o tempo e o erro de cada modelo varia linearmente com o tempo.

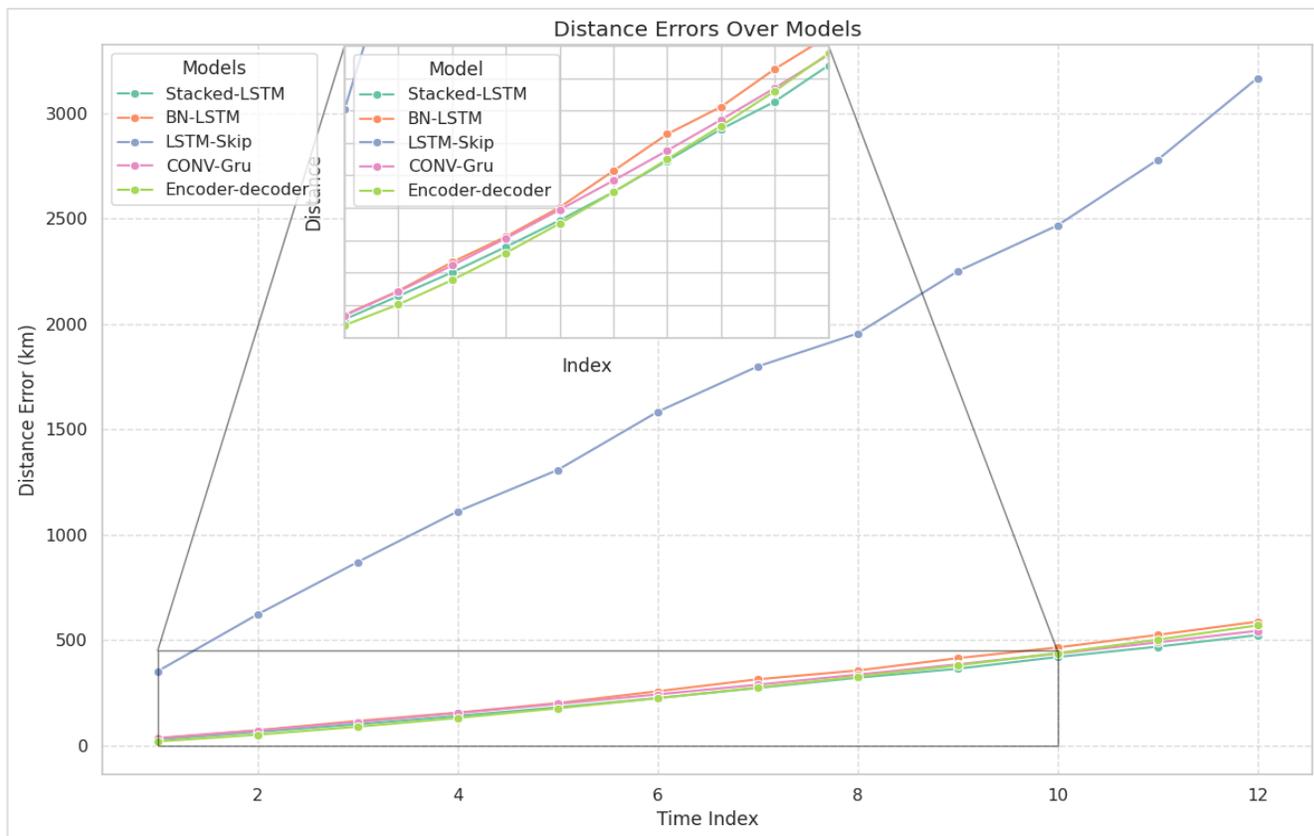


FIGURA 25: VARIAÇÃO DO ERRO MÉDIO DA DISTÂNCIA (ENTRE POSIÇÕES REIS E ESTIMADAS) DOS 5 MODELOS DE REDES NEURAIIS PROPOSTOS NESTE TRABALHO PARA FAZER PREVISÕES DE $(\Delta\text{LAT}, \Delta\text{LON})$ DE 6–72 HORAS (DE 6 EM 6 HORAS): VERDE - STACKED-LSTM, LARANJA - BN-LSTM, AZUL - SKIP-LSTM, VERDE - ENCODER-DECODER

Sob o ponto de vista de estatística descritiva, na presença de casos extremos ou de uma distribuição de dados inclinada, a moda e mediana são mais robustas como medidas de tendência central do que a média (Bruce & Bruce, 2017). Portanto, embora estes modelos tenham se mostrado eficazes de avaliação das medias, para se obter uma descrição mais informada da qualidade dos modelos, é importante também verificar e analisar da distribuição do erro de cada um dos modelos a cada instante da previsão. Neste contexto, a **Figura 26** apresenta a distribuição do erro da distância de cada modelo a cada instante de previsão que foi usada para auxiliar na seleção dos modelos de acordo com a sua performance.

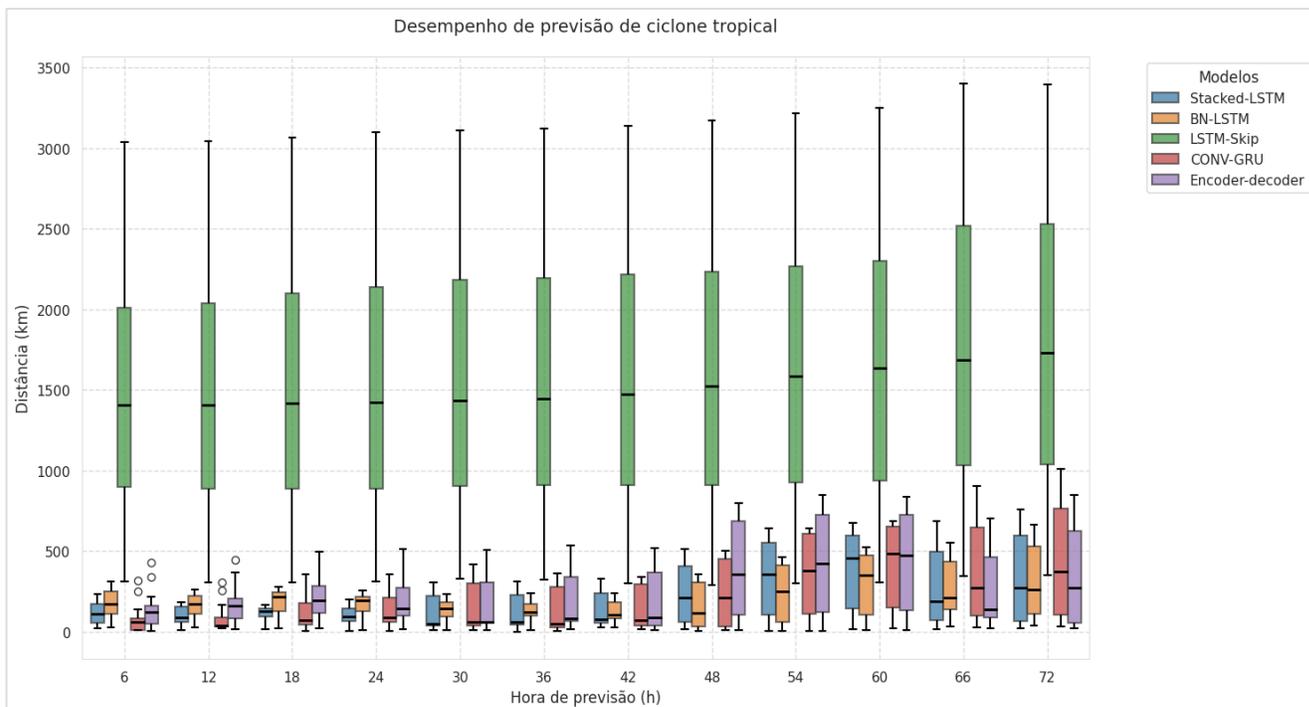


FIGURA 26: DIAGRAMA DE CAIXA DO ERRO MÉDIO DA DISTÂNCIA (ENTRE POSIÇÕES REIS E ESTIMADAS) DOS 5 MODELOS (ARQUITETURAS DE REDES NEURAI) PROPOSTOS NESTE TRABALHO PARA FAZER PREVISÕES DE (δlat , δlon) DE 6–72 HORAS (NUM INTERVALO DE 6 HORAS).

4.3. Seleção de Modelos

Conforme verificado antes nas figuras anteriores, o modelo (arquitetura) LSTM-SKIP não foi eficaz em extrair ou aprender os recursos (características) e padrões de trajetórias necessários para prever, com precisão, a trajetória dos CTs sobre o SOI. Portanto, apresenta um diagrama mais amplo cujos extremos são significativamente maiores, atingindo cerca de $400km$ e $3000km$ de erro mínimo e máximo, respectivamente. Enquanto os valores dos outros modelos concentram-se entre $0km$ a $1000km$. Portanto, o modelo LSTM-SKIP é descartado.

Nas primeiras 24 horas, apesar dos diagramas e amplitude interquartil de todos os modelos serem menores, existe uma ligeira diferença na distribuição do erro de cada um deles. Neste caso, verifica-se que apesar do modelo Stacked-LSTM e ENCODER-DECODER superarem todos os outros modelos, incluindo o Conv-GRU em termos da média (Tabela 11), a Amplitude Interquartil e mediana do Conv-GRU é relativamente menor do que dos outros modelos. Por outro lado, após as primeiras 24 horas, no intervalo de 30-48 horas, verifica-se um padrão ligeiramente similar, embora o modelo BN-LSTM apresente as menores Amplitudes Interquartil. No entanto, embora a mediana e média (Tabela 11) sejam

menores para todos os modelos, a distribuição do erro mostra-se inclinada para regiões de maior erro. No próximo intervalo de previsão verifica-se um padrão ligeiramente similar aos outros intervalos.

Com base nas observações e informações extraídas das **Tabelas 9 e 10**, junto das observações feitas na **Figura 26** discutidas acima e, assumindo que comportamento dos modelos verificado no conjunto de teste se mantenha nos casos futuros, sugere-se que a previsão de trajectória de CTs sobre o SOI seja feita com base nos modelos ENCODER-DECODER, Stacked-LSTM e Conv-GRU.

No entanto, visto que estes modelos apresentam diferentes aspectos positivos de performance, sugere-se que a previsão final seja feita através da combinação (*ensemble*) destes modelos. Neste estudo, realiza-se a combinação destes modelos através do método *ensemble* descrito na secção [3.2.4].

4.4. Produção de Trajectórias

Para demonstrar melhor a efetividade dos modelos seleccionados [4.3.] para a previsão de trajetórias de CTs, os ciclones Kenneth e Eloise foram escolhidos como estudo de caso para a previsão de trajetórias. Os Resultados das previsões foram baseadas nos dados das anteriores 24 horas de cada CT para prever as próximas 72 horas dos mesmos ciclones gerados pelo método de janela [2.3.5] e, a construção das trajetórias é feita conforme descrito na secção [3.2.3]. As **Figura 27**, **Figura 28 e Figura 29** mostram as previsões de trechos das trajetórias dos ciclones Kenneth e Eloise feitas por cada um dos modelos, incluído a trajectória observada, trajectórias do Stacked-LSTM, Encoder-Decoder, Ensemble-Model, Meteo-France e Conv-GRU, conforme o procedimento descrito na secção 3.2.4.

De modo geral, para os primeiros instantes, verifica-se que as previsões dos modelos aproximam-se das posições reais dos CTS. No caso da **Figura 27**, verifica-se que modelo o Metéo-France supera as previsões feitas pelos modelos propostos neste estudo durante uma boa parte do trajecto. No entanto, verifica-se que o modelo Conv-GRU mostra-se consistente durante todo o trajecto apresentando uma aproximação significativa ao trajecto observado. Por outro lado, verifica-se que o Ensemble-Model sofre forte influência dos modelos Encoder-Decoder e Stacked-LSTM, que tendem a se desviar para o sul. No entanto, para os primeiros instantes as previsões dos mesmos mostram-se precisas.

Por outro lado, a **Figura 28** as trajetórias previstas também aproximam-se das trajetórias reais. Mas também é possível verificar que é particularmente difícil para os modelos prever curvaturas repentinas na trajetória dos CTs, é possível também observar que os modelos propostos superaram o modelo do Meteo-France neste caso. A **Figura 29** mostra um padrão similar ao observado na **Figura 27**, onde o modelo do Meteo-France supera, ligeiramente, os demais modelos.

Comparação da Previsão da Trajetória do CT Kenneth sobre SOI

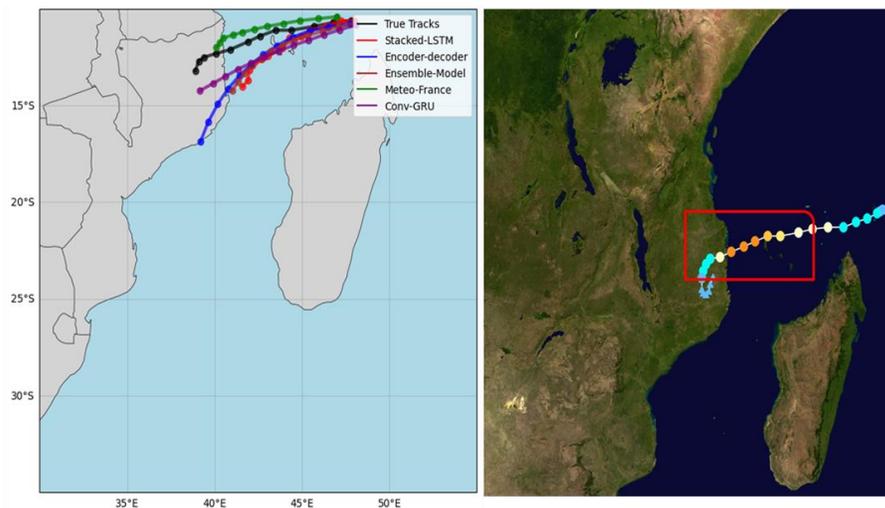


FIGURA 27: MAPA DA TRAJETÓRIA DO CT KENNETH DA TEMPORADA CICLÔNICA DE 2018-2019 SOBRE O SOI.

Na Figura 27, o retângulo vermelho na imagem a esquerda indica o trecho da trajetória observada que deve ser reproduzida pelos modelos; o mapa a direita apresenta o segmento da trajetória observado (a preto) e as trajetórias previstas pelos modelos Stacked-LSTM, Encoder-Decoder, Conv-GRU, Ensemble-Model e Meteo-France representadas pelas cores vermelho, azul, lilás, Castanho e verde, respectivamente

Comparação da Previsão da Trajetória do CT Eloise sobre SOI

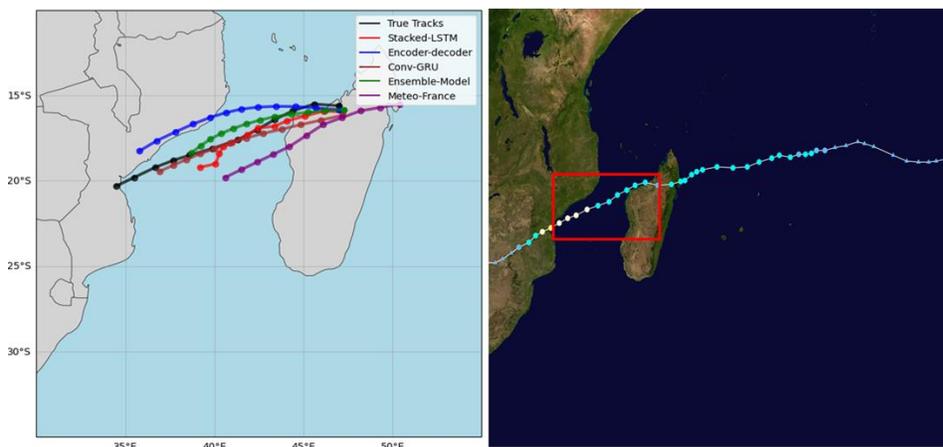


FIGURA 28: MAPA DA TRAJETÓRIA DO CT ELOISE DA TEMPORADA CICLÔNICA DE 2020-2021 SOBRE O SOI. Na Figura 28, o retângulo vermelho na imagem a esquerda indica o trecho da trajetória observada que deve ser reproduzida pelos modelos; o mapa a direita apresenta o segmento da trajetória observado (a

preto) e as trajetórias previstas pelos modelos Stacked-LSTM, Encoder-Decoder, Conv-GRU, Ensemble-Model e Meteo-France representadas pelas cores vermelho, azul, Castanho, verde e lilás, respectivamente.

Comparação da Previsão da Trajectória do CT Eloise sobre SOI

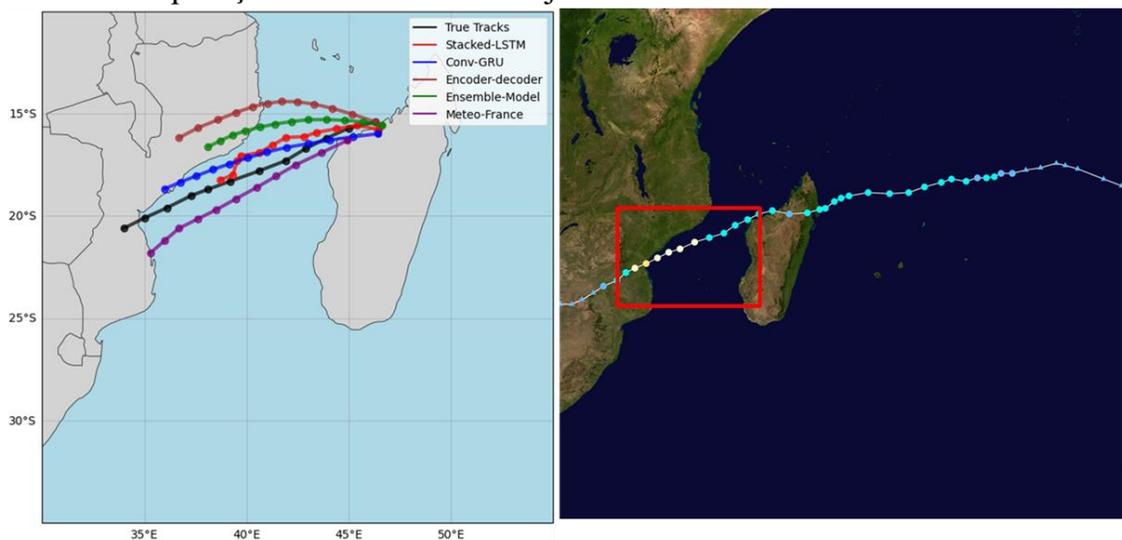


FIGURA 29: MAPA DA TRAJETÓRIA DO CT ELOISE DA TEMPORADA CICLÓNICA DE 2020-2021 SOBRE O SOI. Na Figura 29, o retângulo vermelho na imagem a esquerda indica o trecho da trajetória observada que deve ser reproduzida pelos modelos; o mapa a direita apresenta o segmento da trajetória observado (a preto) e as trajetórias previstas pelos modelos Stacked-LSTM, Encoder-Decoder, Conv-GRU, Ensemble-Model e Meteo-France representadas pelas cores vermelho, azul, Castanho, verde e lilás, respectivamente.

Capítulo 5 : Considerações finais e Recomendações

5.1. Conclusão

Este estudo utilizou o Aprendizado Profundo (Deep Learning, DL) para modelar o movimento de Ciclones Tropicais (CTs) sobre a Bacia do Sudoeste do Oceano Índico (SOI). Cinco modelos baseados em redes neurais recorrentes (STACKED-LSTM, BN-LSTM, SKIP-LSTM, CONV-GRU e ENCODER-DECODER) foram desenvolvidos e treinados para prever as trajetórias dos CTs, utilizando como entrada os dados das últimas 24 horas de informações meteorológicas e de observação de trajetória dos CTs. Antes do treinamento das redes neurais profundas, o algoritmo Random Forest Regressor foi empregado para selecionar as variáveis mais relevantes, com base nos dados extraídos do banco IBTrACS e variáveis derivadas, por meio da análise da importância de cada variável. O algoritmo permitiu a seleção de 11 variáveis (ANG, LON, DIR_DIF, MER_DIS, LON_DIF, LAT, MOV_SP, COR, ZON_DIS, PRE, WIND_SP) que melhor explicam as mudanças de latitude e longitude (δlat , δlon) dos CTs nas próximas 6 a 72 horas.

Após o treinamento, todos os cinco modelos foram avaliados e comparados com base em métricas de erro, incluindo Erro Médio Absoluto (MAE), Raiz do Erro Quadrático Médio (RMSE) e erro de distância nas previsões realizadas com dados de teste. Os resultados sugerem que os modelos **Conv GRU**, **Stacked-LSTM** e **Encoder-Decoder** apresentaram um desempenho superior em relação aos outros modelos, sendo mais eficazes na previsão das trajetórias dos CTs sobre o SOI.

No entanto, a análise das previsões revelou diferenças no desempenho desses modelos dependendo do horizonte de previsão. O **modelo Encoder-Decoder** se destacou nas previsões de curto prazo (até 42 horas), enquanto o **Stacked-LSTM** obteve um desempenho superior nas previsões de médio prazo (42–72 horas). Embora o **Conv-GRU** tenha mostrado um desempenho consistente em termos de dispersão, ele não se destacou tanto quanto os modelos mencionados em termos de precisão.

Com base nas habilidades e características de desempenho de cada modelo, este estudo sugere que a previsão de trajetórias de CTs sobre o SOI seja realizada com os modelos **Encoder-Decoder**, **StackedLSTM** e **Conv-GRU**. No entanto, dado que cada modelo apresenta aspectos positivos distintos, recomenda-se que a previsão final seja feita por meio de uma combinação dos três modelos, aproveitando suas respectivas forças para garantir maior precisão nas previsões.

Por fim, ao comparar o desempenho dos modelos desenvolvidos neste estudo com os modelos atualmente utilizados na região, observa-se um potencial significativo de melhoria nas previsões de trajetórias de

CTs, o que sugere que esses modelos baseados em Aprendizado Profundo podem oferecer uma solução mais eficaz e precisa para previsões operacionais no SOI.

5.2. Recomendações

5.2.1. Recomendações para a Secção do Curso de Meteorologia da UEM

À Secção do Curso de Meteorologia Recomenda-se que:

- No programa de formação do Meteorologista, crie-se a possibilidade de se adicionar uma cadeira de ciência de dados para que os formandos não tenha dificuldades em aplicar as tecnologias usadas em ciência de dados, principalmente sistemas de aprendizagem máquinas e mineração de dados, em suas pesquisas ou nos sectores de trabalho.

5.2.2. Recomendações para Próximas Pesquisas

- No âmbito da aplicação de *Machine Learning* para modelagem de trajectória de CTs, recomenda-se que os próximos estudos possam incorporar a informações meteorológicas do meio ambiente adjacente ao CT nos *inputs*, visto que estes podem influenciar bastante o movimento do sistema.

5.2.3. Recomendações para o INAM

- Recomenda-se ao INAM que invista recursos para o desenvolvimento de modelos baseados em tecnologia de *Machine Learning* para o monitoramento e previsão de fenómenos Meteorológicos, como Tempestades.

Bibliografia

1. Aberson, S. D. (1998). Five-Day Tropical Cyclone Track Forecasts in the North Atlantic Basin. *AMS*, 1005–1015.
2. Alemany, S., Beltran, J., Perez, A., & Ganzfried, S. (2019). Predicting Hurricane Trajectories Using a Recurrent Neural Network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), 468-475.
3. Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and translate. *ArXiv*.
4. Boussioux, L., Zeng, C., Guénais, T., & Bertsimas, D. (2022). Hurricane Forecasting: A Novel Multimodal Machine Learning Framework. *AMSO*, 817-831. Obtido de
5. Brownlee, J. (2019). *Better Deep Learning: Train Faster, Reduce Overfitting and Make better Predictions*. Machine Learning Mastery.
6. Brownlee, J. (2019). *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs, and LSTMs in Python*. Machine Learning Mastery.
7. Bruce, P., & Bruce, A. (2017). *Practical Statistics for Data Scientists*. Sebastopol: O'Reilly.
8. Brunton, S. L., & Kutz, J. N. (2021). *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* (2 ed.). Cambridge: Cambridge University Press.
9. Cangialosi, J. P. (2020). National hurricane center forecast verification report. *National Hurricane*, 77.
10. Chan, J., & Kepert, J. (2010). Global Perspectives on Tropical Cyclones: From Science to Mitigation. *World Scientific Series on Asia-Pacific Weather and Climate*, World Scientific, 448.
11. Chollet, F. (2017). *Deep Learning with Python*. Manning.
12. Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). *Mathematics for Machine Learning* (I ed.). (L. Cowles, Ed.) Cambridge: Cambridge University Press.
13. DeMaria, M., Mainelli, M., K. Shay, L., Knaff, J. A., & Kaplan, J. (2005). Further Improvements to the Statistical Hurricane Intensity Prediction Scheme (SHIPS). *AMS Journals*, 531-543.
14. ECMWF. (2013). 'Who we are'. Obtido em 28 de 05 de 2024, de ECMWF.
15. ECMWF. (2019). IFS documentation Part III: Dynamics and numerical procedures. *ECMWF*, 3, 31.
16. Géron, A. (2019). *Hands-on Machine Learning with Scikit-learning, Keras & Tensorflow*. Canada: O'Reilly.

17. Giffard-Roisin, S., Yang, M., Charpiat, G., Bonfanti, C. K., Kégl, B., & Monteleon, C. (2020). Tropical Cyclone Track Forecasting Using Fused Deep Learning From Aligned Reanalysis Data. *Data Frontiers*, 3(1).
18. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep LEarning*.
19. Hall, T., & Jewson, S. (2007). Statistical modelling of North Atlantic tropical cyclone tracks. *Tellus A Dyn. Meteorol. Oceanogr.*, 486–498.
20. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving Neural Networks by Preventing Co-Adaptation of Features Detectors. *arXiv*. doi:1207.0580v1
21. Holland, G. J. (1983). Tropical cyclone motion: Environmental interaction plus a beta effect. *J. Atmos. Sci.*(40), 328-342.
22. Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate shift . *Proccedings of the 32nd International Conference on machiine Leaning* , 448-456.
23. Kolstad, E. (2021). Prediction and precursors of Idai and 38 other tropical cyclones and storms in the Mozambique Channel. *Q J R Meteorol Soc.*, 147, 45–57.
24. Leroux, M.-D., Meister, J., Mekies, D., & Dorla, A.-L. (2018). A Climatology of Southwest Indian Ocean Tropical Systems: Their Number, Tracks, Impacts, Sizes, Empirical Maximum Potential Intensity, and Intensity Changes. *American Meteorological Society.*, 57, 1021-1040. doi:10.1175/JAMC-D-17-0094.1
25. Moroney, L. (2021). *AI and Machine learning for coders*. Sebastopol: O'Relly.
26. Petterson, J., & Gibson, A. (2017). *Deep Learning: A Practitioner's Approach* . Sebastopol : O'Reilly .
27. Prince, S. J. (2024). *Understanding Deep Learning*.
28. Python Software Foundation. (2019). *Python*.
29. Raschaka, S., & Mirijalili, V. (2019). *Python Machine Leaning* (3 ed.). Birmingham, UK: Packt Publishing Ltd.
30. Simon, A., Penny, A. B., DeMaria, M., Franklin, J. L., Pasch, R. J., Rappaport, E. N., & Zelinsky, D. A. (2018). A Description of the Real-Time HFIP Corrected Consensus Approach (HCCA) for Tropical Cyclone Track and Intensity Guidance. *AMS*, 37–57.

31. Srivastava, N. (2014). Dropout: A simple way to prevent Neural Networks from overfitting. *Journal of Machine Learning Research*, 15, 1929-1958.
32. Stull, R. (2015). *Practical Meteorology: An Algebra-based Survey of Atmospheric Science*. An Algebra-based Survey of Atmospheric Science: An Algebra-based Survey of Atmospheric Science.
33. Tian, W., Lai, L., Niu, X., Zhou, X., Zhang, Y., & Kenny, L. K. (2023). Estimation of Tropical Cyclone Intensity Using Multi-Platform Remote Sensing and Deep Learning with Environmental Field Information. *MDPI*.
34. UNEP-Nairobi Convention. (2015). *The Regional State of the Coast Report: Western*. Nairobi: United Nations Environment Programme/Nairobi Convention Secretaria.
35. Wang, L., Wan, B., Zhou, S., sun, H., & Gao, Z. (2023). Forecasting tropical cyclone tracks in the northwestern Pacific. *Geosci. Model Dev*, 16, 2167-179. Obtido de <https://doi.org/10.5194/gmd-16-2167-2023>
36. Wang, Z., Zhao, J., Huang, H., & Wang, X. (2022). A Review on the Application of Machine Learning Methods in Tropical Cyclone Forecasting. *Geoscience model development*, 16, 2167-2178.
37. Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). *Dive into Deep Learning*. Cambridge University Press.

Anexos

- **Medidas de Performance**

O Error Médio Absoluto (*mean absolute error* (MAE)), Raíz do Erro Medio Quadrado (*root mean squared error* (RMSE)), e Error da Mistância em quilômetros (ED) foram escolhidas como as medidas para comparar a performance dos modelos entre as trajetórias reais e previstas.

O MAE é a media dos valores dos desvios entre todos os valores previstos e os valores reais, calculado pela equação Equação 1. Valores altos do erro representam péssimo modelo.

EQUAÇÃO 1:

$$MAE = \sum_{i=1}^n \frac{|P_i - O_i|}{n}$$

Onde n é o número de registros nos dados de teste, P_i é o valor previsto, e O_i é o valor real.

O RMSE é a raiz quadrada do desvio quadrado entre o valor previsto e valor real e o rácio do número de observações. A formula matemática é representada pela Equação 2.

EQUAÇÃO 2:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (P_i - O_i)^2}{n}}$$

O erro da distância (ED) pode medir a distância entre localização prevista e a localização é prevista em quilômetros. Um baixo valor da distância indica um modelo melhor, e EMD é simplesmente a media de ED. A equação matemática é:

EQUAÇÃO 3:

$$ED = R \times \text{Arcos}(\sin(\text{Lat}_{real}) \times \sin(\text{Lat}_{pred}) \times \cos(\text{Lon}_{real} - \text{Lon}_{pred}) + \cos(\text{Lat}_{real}) \times \cos(\text{Lat}_{pred})) \times \pi/180$$

Onde R é o raio da terra, $(\text{Lat}_{real}, \text{Lon}_{real})$ indica a latitude e longitude da localização real do CT, e $(\text{Lat}_{pred}, \text{Lon}_{pred})$ é a localização real. **Fonte especificada inválida..**

Tabela 12: Pseudocódigo do GD

ALGORITMO DO OPTIMIZADOR GRADIENT DESCENT

1: Hiperparâmetros: Taxa de aprendizagem α , número Total de iterações n_{iter} (*epochs*)

2: Iniciliza w aleatoriamente

3: for $i = 1$ to n_{iter} do

4: selecciona todos os j de $\{1, \dots, n\}$, e Actuliza w de $h_w(x)$ através de:

$$\mathbf{w}_i = \mathbf{w}_i - \alpha \nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{b}) \text{ (EQUAÇÃO 4)}$$

Embora não descrito, \mathbf{b} também pode ser actualizado pelo mesmo procedimento.

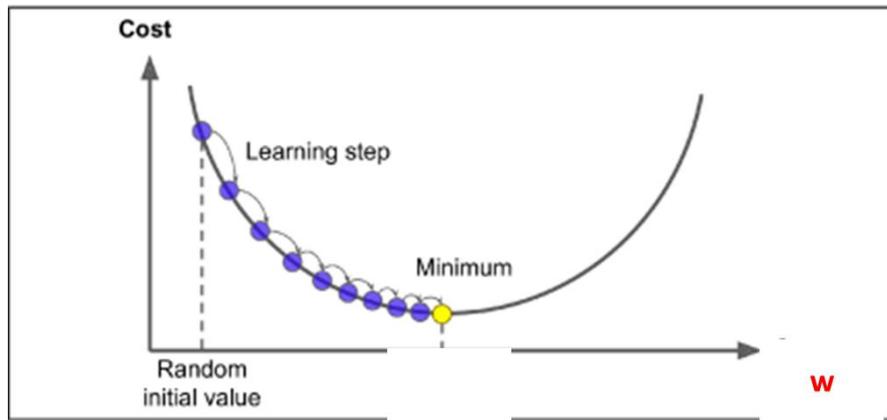


FIGURA 30: VIZUALIZAÇÃO GRÁFICA DO ALGORITMO GD

- **Backpropagation**

Backpropagation usa *gradient descent* para actualizar os parâmetros (\mathbf{w}) de redes neurais artificiais. *Gradient descent* identifica a direção dos pesos que guiam para a maior redução do erro feito pela rede durante as previsões através do gradiente da do erro (ou função de custo) em respeito a cada um dos (\mathbf{w}) do modelo $\nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{b})$. Por outro lado, *Backpropagation* é procedimento para calcular $\nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{b})$ em redes profundas com respeito a cada parâmetro da rede.